(NASA-CR-120308) CREW/COMPUTER
COMMUNICATIONS STUDY. VOLUME 2:
APPENDIXES (McDonnell-Douglas Astronautics
Co.) 130 p HC

N74-32634

CSCL 09B

G3/08   Unclas
          16799

Vol I - N74-18849.

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY

MCDONNELL DOUGLAS
CORPORATION

PRICES SUBJECT TO CHANGE

130

MCDONNELL DOUGLAS

CREW/COMPUTER COMMUNICATIONS STUDY

VOLUME II
Appendixes

FEBRUARY 1974

MDC G5312

J. D. JOHANNES
PRINCIPAL INVESTIGATOR

R. R. JOSLYN
PROJECT MANAGER

CONTENTS

FIGURES

TABLES

GLOSSARY

A/D	Analog to digital
ALC	Automatic level control circuitry
ASCII	American Standard codes for information interchange
A_G	Gross amplitude
A_H	High-frequency amplitude
A_L	Low-frequency amplitude
A_{VH}	Very-high-frequency amplitude
BITE	Built-in test equipment
C/CC	Crew/computer communications
CDH	Constant delta height
CPU	Central processing unit
CRT	Cathode ray tube
CSI	Concentric sequence initiation
CSM	Command service module
D/A	Digital to analog
DEC	Digital Equipment Corporation
DEC-339	CRT display
DR 11-C	Serial interface card
FET	Field effect transistor
FPE	Functional program element
HP/LP	High pass/low pass
IC	Integrated circuit

LED	Light-emitting diodes
LA30	Printer/keyboard
LM	Lunar module
PDP 9	DEC computer which supports the PKD
PDP 11/40	Speech processor computer
PKD	Programmable keyboard and display
REQ Lines	The interrupts
RC	Resistance capacitance
TPF	Terminal phase finalization
TPI	Terminal phase initiation
TTY	Teletypewriter
7700A	Lear Siegler Inc. display

APPENDIX A
PROGRAM DESCRIPTIONS AND USER'S
GUIDE — PART I, PHASE C

Appendix A
PROGRAM DESCRIPTIONS AND USER'S GUIDE
PART I, PHASE C

The following section describes the software routines developed during the Crew/Computer Communications Study. A standardized program description sheet was developed to provide the user with a clear understanding of each routine, any restrictions in its use, the required input data to each routine, and the results after executing the routine. Section A.2 describes how these routines are combined to generate a crew/computer communications application.

A. 1 PROGRAM DESCRIPTIONS

The software has been separated into four groups: system software, application software, on-line utility routines, and off-line utility routines. These four groups contain the following routines:

A.   System Software
     PKD — PKD handler

B.   Application Software
     MAIN — Application executive for C/CC demonstration
     LNK1 — Core overlay No. 1
     LNK2 — Core overlay No. 2
     RENCAL—Rendezvous calculation routine

C.   On-line Utility Software
     1.   PKD Handler Interface
          PKIN — Response routine for PKD input
          PKOUT — Initialization routine for PKD handler
     2.   PKD Display Command Routines
          PKCOUT —Character display routine
          PKSOUT— String display routine
          PKPOUT— Page display routine
          PKCL— Cursor location routine
          PKCC— Cursor command routine

1

PKMSL — Special function key light command routine

TIMCON — Time conversion routine (real→ASCII)

TUPDAT — Dynamic time update routine

BLNKY — Display key blanking routine

GETPKD — Display data retrieval routine

3. Keyboard Response Routines

NUMIN — Numeric keyboard response routine (keyboard active)

NUMKEY — Numeric keyboard response routine (keyboard inactive)

MODKEY — Special function keyboard response routine

4. ASCII/Real/ASCII Conversion

KFPA — Floating point to ASCII conversion routine

KAFP — ASCII to floating point conversion routine

5. String Manipulation

PUTC — Single character merge routine

GETC — Single character selection routine

MOVSTR — String merging routine

6. DEC-339 Display Support

GET 339 — Display retrieval routine

7. Error Handling

EROUT 1 — Illegal word error routine

EROUT 2 — Illegal selection error routine

D. Off-line Utility Software

GENPKD — PKD display generation and editing routine

GEN 339 — DEC-339 display page generation routine

GENFIL — Execution table generation routine

CHKPK — Display file checking routine (PKD)

CHK 339 — Display file checking routine (DEC - 339)

WRTPKD — Hard-copy printing routine for PKD displays

CHKOUT — PKD checkout routine

A. 1. 1 System Software

To interface the programmable keyboard display with PDP-9, a PKD
handling routine was added to the system software. Since this routine is not
called by in-line code but is instead called by the monitor or as the result
of an external interrupt, only a description of its operation is included. The

PKD handler performs all communications involving PKD data transfer through the PDP-9 input-output (I/O) channel.

A.1.1.1 Interface with the Monitor

The PKD utility routines communicate with the handler through the I/O macro programs. Whenever an I/O system macro call is executed, control passes through the monitor to the PKD handler. The handler has only one entry point, but contains all the macros for PKD I/O. The first part of this handler discovers which implemented function was requested and transfers control to the corresponding function routine, but ignores requests for nonimplemented functions. When execution of an implemented function routine is completed, control is passed back to the user via the monitor.

To allow better understanding of the handler, the macro function routines are described individually in the following paragraphs.

A.1.1.2 Input-Output System Macro Function Routines

A. .INIT — The program unit executing .INIT associates the handler PKD. with input or output and with a restart address. When the handler is accessed by the monitor for this function, it must provide the monitor with a maximum buffer size, and sets of interrupt channel numbers and associated interrupt routine locations as applicable. Here, subroutines PKIN and PKOUT each call the .INIT function for input and output, respectively.

B. .READ — The program unit executing .READ requests a handler for a device to input a specified number of words into a specified buffer. When the handler is accessed by the monitor to satisfy this request, it will defer the request if input is already underway; otherwise, it will initiate the process. It remains for the associated interrupt routine to carry it out. Here, PKIN calls the .READ function, asking for one word, which is all that the PKD hardware package provides as a result of any human action. PKD input is right-justified in second buffer word (8 bits).

C. .WRITE — The program unit executing .WRITE requests a handler to output to a device a number of 8-bit characters derived from a specified buffer, the exact number being in the buffer header.

3

When the handler is accessed by the monitor to satisfy this request, it will defer the request if output is already underway, and otherwise will initiate the process by sending out the first character. It is up to the associated output complete interrupt routine to complete the process. Here, all of the output subroutines call the .WRITE function.

D.   .WAITR — The program unit executing .WAITR requests a handler to transfer control to a specified location in that program if any process initiated by that handler is not completed, and otherwise to allow execution to continue in sequence. The PKIN subroutine uses .WAITR to determine if its previous .READ request has been satisfied; i.e., has any relevant human action occurred? In either case, it subsequently exits to the application program with at least this status information, allowing the application program to pursue its other activities between frequent looks for a "hit."

The output subroutines do not return status; they merely use the automatic delay-until-finished-with-previous request feature.

E.   .CLOSE — The program unit executing .CLOSE tells a handler that it is (temporarily) through with handler. Here, this is only used and meaningful as part of the restart named by .INIT.

A.1.1.3   Interrupt Routines

The first part of the PKD interrupt routine is exactly the same as for any other device connected to the PDP-9 having its own interrupt channel(s), and communicating under direct program control; i.e., not via a data channel or direct memory access. The first task is to determine which device flag led to the interrupt. Then, the routine will pursue one of two possible paths: one for input and the other for output. It concludes by returning control to the interrupted program.

A.   Input Interrupt Routine — This process consists basically of transferring a character from the associated device (PKD) to the appropriate buffer. When input from the device is complete (after each input character), a specified header must be generated for the buffer. One part of it is the number of word pairs read plus one (the header being a word pair). As many as six data modes

4

might be involved (here, there is only one, Image Alphanumeric; i. e., the data exactly as received, right-justified with leading zeros). Thus, the number of word pairs is always two. Finally, the Input Underway flag is reset. This flag is tested by each execution of .WAITR for input.

B.   Output Interrupt Routine — Character output is accomplished with this routine. After receipt of each complete interrupt output, another character output is initiated, the process being repeated until the requested number of characters have been sent. The number of characters is obtained from the specified output buffer header.

A.1.2 Application Software

The application executive for the Crew/Computer Communications Study demonstration is composed of three software modules: MAIN and two core overlays (LNK1 and LNK2).

The capabilities of the PKD and DEC -339 display terminal were combined in routine RDEM4. Due to the duplication of display data by the CRT and PKD, the development of this routine was terminated before checkout was finished. However, both the CRT and PKD interactive code have been demonstrated independently.

The application-peculiar computation program RENCAL was developed to determine the thrusting requirements for all five LM/CSM rendezvous maneuvers on the Apollo program.

These routines are described in the following text.

NAME: MAIN

PURPOSE: This routine is the application executive of the crew/
                          computer communications demonstration. It handles
                          all the generalized displays and calls the core overlays
                          if they are required.

METHOD: The PKD is polled to establish if a button has been pressed; if it has, the result is checked for errors and if no errors are detected, the execution table for the active display on the PKD is used to establish the appropriate action. If an error is detected, it will be ignored and the PKD will again be polled for crew input response.

RESTRICTIONS: This is a main program and not a callable subroutine.

COMMON: ORA, TRANG, RANGE, RANGRT, YCSI, YDCSI, DELH, TTPI, TCSI, TPC, TCDH, TTPF, ELEV, DVXCSI, DVYCSI, DVZCSI, DVTCSI, DVXPC, DVYPC, DVZPC, DVTPC, DVXCDH, DVYCDH, DVZCDH, DVTCDH, DVXTPI, DVYTPI, DVZIPI, DVTIPI, DVXTPF, DVYTPF, DVZTPF, DVTIPF, ITY, IVA.

SUBROUTINES USED: PKOUT—Initialization routine for PKD handler
                   PKCC—Cursor command routine
                   PKIN—Response routine for PKD input
                   MODKEY—Special function keyboard response routine
                   NUMKEY—Numeric keyboard response routine
                            (keyboard inactive)
                   BLNKY—Display key blanking routine
                   EROUT2—Illegal selection error routine
                   GETPKD—Display data retrieval routine
                   LNK1 —Core overlay No. 1
                   LNK2—Core overlay No. 2

NOTES: LNK1 and LNK2 are core overlays for this application executive. See subroutine RENCAL for COMMON variable definitions.

NAME: LNK1

PURPOSE: This overlay module supplies the programming code required to support the following three rendez-vous maneuvers, CSI, PC, and CDH.

METHOD: The "Chain and Execute" PDP-9 software package is used to generate and execute core overlays, thereby allowing programs to operate independently of core size restrictions. This overlay is used to display the results of rendezvous calculations.

CALL: CALL LNK1

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: ORA, TRANG, RANGE, RANGRT, YCSI, YDCSI, DELH, TTPI, TCSI, TPC, TCDH, TTPF, ELEV, DVXCSI, DVYCSI, DVZCSI, DVTCSI, DVXPC, DVYPC, DVZPC, DVTPC, DVXCDH, DVYCDH, DVZCDH, DVTCDH, DVXTPI, DVYTPI, DVZIPI, DVTIPI, DVXTPF, DVYTPF, DVZTPF, DVTIPF, ITY, IVA

SUBROUTINES USED: KFPA—Floating point to ASCII conversion routine.
TIMCON—Convert real seconds to ASCII hours, minutes, and seconds.
MOVSTR—Merge two ASCII strings.
PKPOUT—Page output routine.

NOTES: All data base communication is accomplished via unlabeled COMMON.

See subroutine RENCAL for COMMON variable
definitions.

EXAMPLE:	None

NAME: LNK2

PURPOSE: This overlay module supplies the programming code required to support the following two rendezvous maneuvers, TPI and TPF. In addition, the closed subroutine RENCAL is included in this overlay and is used to calculate the rendezvous data.

METHOD: PDP-9 software routine "Chain and Execute" is used to generate and execute core overlays. RENCAL is called during the TPI execution and is used to update the data base.

CALL: CALL LNK2

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: ORA, TRANG, RANGE, RANGRT, YCSI, YDCSI, DELH, TTPI, TCSI, TPC, TCDH, TIPF, ELEV, DVXCSI, DVYCSI, DVZCSI, DVTCSI, DVXPC, DVYPC, DVZPC, DVTPC, DVXCDH, DVYCDH, DVZCDH, DVTCDH, DVXTPI, DVYTPI, DVZTPI, DVTTPI, DVXTPF, DVYTPF, DVZIPF, DVTTPF, ITY, IVA

SUBROUTINES USED: KFPA—Floating point to ASCII conversion.
MOVSTR—Merge two ASCII strings.

TIMCON—Convert seconds to ASCI hours, minutes, and seconds.

PKPOUT—Page output routine.

RENCAL—Calculate rendezvous data.

NOTES: All data base communication is accomplished via
                    unlabeled COMMON.

                    See subroutine RENCAL for COMMON variable
                    definitions.

EXAMPLE: None

NAME: RENCAL

PURPOSE: Demonstrate a technique for determining the
                    thrusting requirements for an LM/CSM rendezvous.

METHOD: Clohessy-Wiltshire equations are used to determine
                    the incremental change in velocity (delta) require-
                    ments for each of the five rendezvous burns.

CALL: CALL RENCAL

RESTRICTIONS: The orbits of the active and passive vehicles are
                    assumed to be circular.

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: ORA — Orbital radius of active vehicle (feet)
                    TRANG — Transfer angle between TPI and TPF
                            maneuvers (degrees)
                    RANGE — Range between active and passive vehicle
                            (miles)

RANGRT — Range rate between active and passive vehicle (fps)

YCSI — Out-of-plane position at CSI (miles)

YDCSI — Out-of-plane velocity at CSI (fps)

DELH — Incremental change in height during constant delta height phase (miles)

TTPI — GET time of TPI maneuver (seconds)

TCSI — GET time of CSI maneuver (seconds)

TPC — GET time of plane change maneuver (seconds)

TCDH — GET time of CDH maneuver (seconds)

TTPF — GET time of TPF maneuver (seconds)

PELEV — Elevation angle of passive vehicle (degrees)

DVXCSI — X-axis delta velocity required for CSI maneuver (fps)

DVYCSI — Y-axis delta velocity required for CSI maneuver (fps)

DVZCSI — Z-axis delta velocity required for CSI maneuver (fps)

DVTCSI — Total delta velocity required for CSI maneuver (fps)

DVXPC — X-axis delta velocity required for PC maneuver (fps)

DVYPC — Y-axis delta velocity required for PC maneuver (fps)

DVZPC — Z-axis delta velocity required for PC maneuver (fps)

DVTPC — Total delta velocity required for PC maneuver (fps)

DVXCDH — X-axis delta velocity required for CDH maneuver (fps)

DVYCDH — Y-axis delta velocity required for CDH maneuver (fps)

DVZCDH — Z-axis delta velocity required for CDH maneuver (fps)

DVTCDH — Total delta velocity required for
CDH maneuver (fps)

DVXTPI — X-axis delta velocity required for
TPI maneuver (fps)

DVYTPI — Y-axis delta velocity required for
TPI maneuver (fps)

DVZTPI — Z-axis delta velocity required for
TPI maneuver (fps)

DVTTPI — Total delta velocity required for
TPI maneuver (fps)

DVXTPF — X-axis delta velocity required for
TPF maneuver (fps)

DVYTPF — Y-axis delta velocity required for
TPF maneuver (fps)

DVZTPF — Z-axis delta velocity required for
TPF maneuver (fps)

DVTTPF — Total delta velocity required for
TPF maneuver (fps)

ITY — PKD keyboard type flag

IVA — PKD keyboard value flag

SUBROUTINE USED: NUMIN — Numeric keyboard input routine

KAFP — ASCII to floating point conversion routine

NOTES: TPI time and elevation angle are loaded at the
numeric keyboard during the execution of this
routine.

The following initial conditions are assumed for
the program:

a) CSI time — 101:21.42 (GMT)
b) Lunar orbit
c) Initial orbit height of active vehicle = 25.0 (miles)
d) Initial orbit height of passive vehicle = 60.0 (miles)

11

e) X position at CSI = 160. 0 (miles), passive vehicle-centered coordinate system

f) Y position at CSI = 0.174 (miles), passive vehicle-centered coordinate system

g) Z position at CSI = 35. 0 (miles), passive vehicle-centered coordinate system

h) X-velocity at CSI = 88.25 (fps), passive vehicle-centered coordinate system

i) Y-velocity at CSI = 0. 065 (fps), passive vehicle-centered coordinate system

j) Z-velocity at CSI = 0. 0 (fps), passive vehicle-centered coordinate system

A. 1. 3 On-Line Utility Software

The support routines that would be used during the execution of an interactive application program are described in this section. They are organized as follows:

A. PKD Handle Interface

B. PKD Display Commands

C. Keyboard Response

D. ASCII/ REAL/ ASCII Conversion

E. String Manipulation

F. DEC-339 Display Support

G. Error Handling

A. 1. 4 PKD Handler Interface Routines

Two routines fall in this category. The first, PKOUT, is used to initialize the handler for output to the PKD while the second, PKIN, is used to respond to computer input from the PKD.

It should be noted that the user program is not (logically) interrupted when a PKD button is pressed. Pushbutton action merely causes an indicator to be set in PKIN, and this may affect the user's flow of control at his discretion.

NAME:	PKIN

PURPOSE: Informs if any new input from the PKD hardware package has been received, and if so, what type it is, what value for that type, and, if applicable, provides an internally usable form (IOPS ASCII) of that value.

METHOD: Uses .READ to check for input. Argument values are set according to whether an input was received or not. When an input is received, the response is analyzed and the arguments set.

CALL: CALL PKIN (status flag, type, value, internal value, error code).

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS:
A. Status flag: = 1 if new input received,
     = 0 otherwise

B. Type: Most significant four bits from PKD, plus one

   = 1 if display matrix switch
   = 2 if mode control switch
   = 3 if numeric keyboard

   Other values are not wired; therefore, they are illegal if received

C.   Value: Least significant four bits from PKD,
plus one

   = integer 1-16 if Type 1
   = integer 1-12 if Type 2
   = integer 1-12 if Type 3

Other values are not wired; therefore,
they are illegal for Type 2 or 3, and
meaningless otherwise.

D.   Internal value: IPOS ASCII (PDP-9 internal
ASCII) equivalent of numeric keyboard
input, otherwise zero.   At least two
words must be provided for this variable.
Value is left justified in first word,
remainder of first and all of second word
set to zero.

E.   Error code: = 0 if none detected
   = 1 for Type 2 value out of range
   = 2 for Type 3 value out of range
   = 3 if Type 4 through 9
   = 4 if Type greater than 9 (16 is
      max type possible)

COMMON: None

SUBROUTINES USED: .INIT, .READ, .WAIT, .CLOSE, .EXIT

NOTES: All of the arguments are results; hence, they must
be coded as variables of the using FORTRAN applica-
tion program.

EXAMPLE: None

14

NAME: PKOUT

PURPOSE: Initializes PKD handler for output and must be
                         called just once, before any other output subroutine
                         is called.

METHOD: CALLS .INIT for output and establishes address for
                         .CLOSE and .EXIT for restart.

CALL: CALL PKOUT

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: .INIT, .CLOSE, .EXIT

NOTES: PDP-9 monitor set up to notify PKD handler of
                         complete interrupt outputs from LT-19.

EXAMPLE: None

A.1.4.1 PKD Display Command Routines

Ten routines have been developed to support the programmable command
keyboard display. The functions performed by these routines include:
display characters and strings, cursor control commands, changing the
state of the special function keyboard lights, displaying time, blanking
display data in a programmable command keyboard switch location, and
retrieving data from the disk.

15

NAME: PKCOUT

PURPOSE: Issues command to display specified character at
                        current cursor location.

METHOD: The character is sent using . WRITE

CALL: CALL PKCOUT (character)

RESTRICTIONS: None

INPUT ARGS: Character—Single character in IOPS ASCII formatted
                        variable (left justified).

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: . WRITE

NOTES: None

EXAMPLE: Display a "1" at current cursor location.
                        DATA A/5H1 .... /
                           :
                           :
                        CALL PKCOUT (A)
                           :
                           :

NAME: PKSOUT

PURPOSE: Issues commands to display specified string of char-
                        acters, starting at current cursor location. The
                        subroutine returns to the user after issuing the first
                        command, the remainder to be issued as a

consequence of receiving complete interrupt outputs. Thus, the application program can continue concurrent with the display update process.

METHOD: The character string is supplied to .WRITE, which outputs the number of characters specified.

CALL: CALL PKSOUT (character array, number of characters)

RESTRICTIONS: None

INPUT ARGS:
A. Character array—Real array, first word pair of which is reserved for use by .WRITE, character string starting at second word pair.
B. Number of characters—Character count, meaningful only for 1 to 255.

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: .WRITE

NOTES: None

EXAMPLE: Display numbers 1, 2, 3, 4, 5 starting at present cursor location.
Dimension A(2)
DATA A(2)/5H12345/
:
:
ICNT=5
CALL PKSOUT (A(1), ICNT)
:
:

NAME: PKPOUT

PURPOSE: Issues command to home cursor. Then returns con-
                        trol to user having arranged to issue commands to
                        display the specified page of 255 characters in con-
                        junction with receipt of complete interrupt output.
                        Concurrency of the application program with the
                        display process becomes quite important at the page
                        level, for with the PKD connected to the PDP-9 via a
                        telecommunications link, it takes just over two
                        seconds to put up a whole new page.

METHOD: Calls PKCC with a home command, then calls PKSOUT
                        to output 255 characters.

CALL: CALL PKPOUT (Character array)

RESTRICTIONS: None

INPUT ARGS: Character array—Real array of size 52, first word
                        pair reserved for use by PKSOUT, remaining 51
                        word pairs contain the ASCII characters for display.

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKCC, PKSOUT

NOTES: None

EXAMPLE: Display ASCII data stored in array PAGE 1 on the PKD.
                           :
                           :
                        CALL PKPOUT (PAGE 1(1))
                           :
                           :

18

NAME: PKCL

PURPOSE: To position the displays cursor, which establishes
                            start of character display.

METHOD: Calls .WRITE to issue first of two commands
                            required to position the cursor at a specified loca-
                            tion, then returns to user having arranged to send
                            the second command upon receipt of the next com-
                            plete interrupt output.   The location is specified by
                            row and column.   The PKD display being treated as an
                            8 x 32 matrix.   Row is integer $0 - 7$, column $0 - 31$.

CALL: CALL PKCL (row position, column position)

RESTRICTIONS: None

INPUT ARGS: Row position — integer $0 - 7$
                            Column position — integer $0 - 31$

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: . WRITE

NOTES: None

EXAMPLE: :
                            IROW = 1
                            ICOL = 3
                            CALL PKCL (IROW, ICOL)
                                :
                                :

NAME: PKCC

PURPOSE: Issues command to effect specified control of cursor
                        and also clearing display if requested.

METHOD: Calls . WRITE to issue specified command.

CALL: CALL PKCC (cursor control code)

RESTRICTIONS: None

INPUT ARGS: Cursor control code: = 1 for clear display (and
                                                 home cursor)
                                             = 2 for home cursor
                                             = 3 for visual cursor ON
                                             = 4 for visual cursor OFF
                                             = 5 for cursor blink ON
                                             = 6 for cursor blink OFF
                                             = 7 for new line
                                             = 8 (currently undefined)
                                             = 9 for cursor right
                                             = 10 for cursor left
                                             = 12 for cursor down

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: . WRITE

NOTES: Clear display takes almost 0.5 millisecond, before
                        which no further commands will be accepted by the
                        display. (Note that page output to display is preceded
                        only by home cursor.) The user may first request
                        clear display, arrange for the requisite delay, then

use either PKPOUT or PKSOUT to output only a
portion of a page, if the remainder of that page is
to be blank.

EXAMPLE: Turn cursor OFF
                        :
                        :
                     ICMD = 4
                     CALL PKCC (ICMD)
                        :
                        :

NAME: PKMSL

PURPOSE: Issues command to set specified mode switch light to
                     specified state (ON or OFF).

METHOD: The output word is formed and dispatched with
                     .WRITE.

CALL: CALL PKMSL (mode switch code, desired state)

RESTRICTIONS: None

INPUT ARGS: A. Mode switch code: integer between 1 and 12,
                         inclusive
                     B. Desired state: = 0 for OFF
                                         = 1 for ON

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: .WRITE

NOTES: None

EXAMPLE: Turn mode control light 10 "ON"
                        :
                        :
                   IMSWT = 10
                   IOSWT = 1
                   CALL PKMSL (IMSWT, IOSWT)
                        :
                        :

NAME: TIMCON

PURPOSE: Convert time in seconds to three ASCII strings for
                   displaying hours, minutes, and seconds.

METHOD: Divide seconds by 3,600 and convert to integer to
                   get hours, multiply fractional result of above by 60,
                   and convert to integer to get minutes.   Take frac-
                   tional result of previous calculation and multiply by
                   60 to get seconds.

                   Float the above three integers and use subroutine
                   KFPA to convert floating point number to ASCII data.

CALL: CALL TIMCON (TSEC, AHR, AMIN, ASEC)

RESTRICTIONS: None

INPUT ARGS: TSEC — Time in seconds (real number)

OUTPUT ARGS: AHR — Three word ASCII image of hours
                   AMIN — Three word ASCII image of minutes
                   ASEC — Three word ASCII image of seconds

COMMON: None

SUBROUTINES USED: KFPA — Floating point to ASCII conversion

NOTES: First word of ASCII array is filled in by display routine PKSOUT.

EXAMPLE: Convert 3,605 seconds to ASCII hours, minutes, and seconds.

Dimension AHR (3), AMIN (3), ASEC (3)

    :

    :

TSEC = 3605.0

CALL TIMCON (TSEC, AHR (2), AMIN (2), ASEC (2))

    :

    :

NAME: TUPDAT

PURPOSE: Dynamically update time displayed on PKD.

Note: Time is a integer number while the same number on the display must be in ASCII representation.

METHOD: Set index pointers to low-digit value of initial minute and second value. Use pointer to index ASCII data every second and incremental minutes every 60 seconds. Results are displayed at requested location on PKD.

CALL: CALL TUPDAT (MIN, ISCC, IROW, ICOL)

RESTRICTIONS: No hours are shown, so minutes continue incrementing after reaching a value of 59. Maximum value of minutes is 99.

INPUT ARGS:	MIN — Initial value of minutes
	ISEC — Initial value of seconds
	IROW — Start location of display data (row)
	ICOL — Start location of display data (column)
OUTPUT ARGS:	None
COMMON:	None
SUBROUTINES USED:	PKIN — Response routine for PKD
	PKCL — PKD cursor location
	PKCOUT — PKD character display
NOTES:	
EXAMPLE:	Start incrementing clock set to "0" and display at cursor location 4, 10 (row, column).

```
    :
    :
MIN = 0
ISCC = 0
IROW = 4
ICOL = 10
CALL TUPDAT (MIN, ISCC, IROW, ICOL)
    :
    :
```

NAME:	BLNKY
PURPOSE:	Blank the 16-character display positions under the selected command display key.
METHOD:	Fill in blank characters in display string.
CALL:	CALL BLNKY (MKEY)

RESTRICTIONS: None

INPUT ARGS: MKEY — Command key to be blanked.

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKCL — Cursor location command routine
                       PKCOUT — Character output routine

NOTES: The reference display file stored on the disk is not
                       modified.

EXAMPLE: Blank the depressed command display key.
                              :
                              :
                    10 CALL PKIN (IST, ITY, IVAL, IIV, IER)
                       IF (IST. EQ. 0) GO to 10
                       IF (ITY. EQ. 1) GO to 20
                              :
                              :
                    20 CALL BLNKY (IVAL)
                              :
                              :

NAME: GETPKD

PURPOSE: Get requested PKD display file data from disk.

METHOD: Read requested display file from disk.

CALL: CALL GETPKD (PKDF, RPKD)

RESTRICTIONS: IOPS 13 results if requested file is not found on disk

INPUT ARGS: PKDF — Display file name

OUTPUT ARGS: RPKD — Display data (53 word array)

COMMON: None

SUBROUTINES USED: None

NOTES: None

EXAMPLE: Read from disk and store the display data for FILE 1 in array RPKD.

Data PKDF/5HFILE1/

        :

        :

Call GETPKD (PKDF, RPKD)

        :

        :

A.1.4.2 Keyboard Response Routines

In-line code in main and core overlays is used to respond to the programmable command keyboard while FORTRAN — callable subroutines are used to respond to the special function keyboard and the numeric keyboard. The following routines allow the user to respond to the latter two keyboards.

NAME: NUMIN

PURPOSE: Read numbers from numeric keyboard and store characters in array NUM. Numbers are displayed on command keyboard display as they are received.

METHOD: Use ASCII characters as supplied by PKIN and transfer left justified into array NUM (1 character/array element).

CALL: Call NUMIN (IR, IC, ICCNT, NUM, ITY, IVAL)

RESTRICTIONS: None

INPUT ARGS: IR — Row location of first input character

                         IC — Column location of first input character

                         ICCNT — Total number of input characters

OUTPUT ARGS: NUM — Array containing ASCII data

                         ITY — Keyboard type flag

                         IVAL — Key number flag

COMMON: None

SUBROUTINES USED: PKCL — Cursor positioning routine

                         PKCC — Cursor control command routine

                         PKIN — PKD response routine

                         PKCOUT — Character output routine

NOTES: All three keyboards are active when routine NUMIN
                         is waiting for numeric input. It is therefore possible
                         to leave this routine in the middle of a data entry
                         operation.

EXAMPLE: Read three numeric values into array NUM and
                         display at location 4, 10 (row, column)

                             :

                         IR = 4

                         IC = 10

                         ICCNT = 3

                         CALL NUMIN (IR, IC, ICCNT, NUM, ITY, IVAL)

                             :

                             :

NAME: NUMKEY

PURPOSE: Respond to numeric keyboard.

METHOD: Issue message: "Error: Activity not programmed"

CALL: CALL NUMKEY

RESTRICTIONS: See NOTES

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKCL — Cursor location command
                      EROUT2 —Error message routine

NOTES: Routine NUMIN will be merged into this routine. It
                      will, however, be necessary to conduct a test to
                      verify that the numeric keyboard is active, and if not,
                      to display the above error message. At present, the
                      same result could be achieved by calling EROUT2
                      with the input argument set to +1.

EXAMPLE: None

NAME: MODKEY

PURPOSE: Response routine for the 12 special function keys.

METHOD: Set pointers, set indexes, or display messages as
                      required.

CALL: CALL MODKEY (IVA, FIL, ARRAY, MAX)

28

RESTRICTIONS: Only MONITOR, BACKUP, and EMERGENCY STOP
                    are implemented; all other key depressions result
                    in the following message being displayed: "ERROR:
                    ACTIVITY NOT PROGRAMMED"

INPUT ARGS: IVAL — Special function key number

OUTPUT ARGS: FIL — Modified file directory name for new response
                        data
                    ARRAY — Trail of file director names from top level
                        to present position
                    MAX — Pointer to array element

COMMON: None

SUBROUTINES USED: PKCL — Cursor location command routine
                    PKSOUT — String output to PKD display
                    EROUT2 — Error message routine

NOTES: None

EXAMPLE: Respond to selected special function key
                        :
                 C READ KEYBOARD HITS.
                10 CALL PKIN (IST, ITY, IVAL, IIV, IER)
                    IF (IST. EQ. 0) GO TO 10
                    IF (ITY. EQ. 2) GO TO 20
                        :

                        :
                20 CALL MODKEY (IVAL, FIL, ARRAY, MAX)
                        :

A.1.4.3 ASCII/REAL/ASCII Conversion

Numeric output from the PKD is in ASCII format. While this format is
necessary to display data, it is meaningless in mathematical operations.

29

The format required for these operations must be either an integer or real number. Two routines were developed to convert from one format to the other.

NAME: KFPA

PURPOSE: Converts a FORTRAN format floating point number
                         to an ASCII character string.

METHOD: The sign character is established and the sign bit
                         discarded. The number is adjusted for a zero
                         exponent. Integers are extracted from the whole
                         number part by division by 10 and from the fractional
                         part by multiplication by 10. As each integer is
                         extracted, it is converted to ASCII format and stored
                         as an individual character. The characters are then
                         assembled into ASCII double words. Unused char-
                         acters in the last word pair are set to blanks.

CALL: CALL: KFPA (real number, ARRAY, W, D)

RESTRICTIONS: Floating point number range: 0.00001 to 131,071.

INPUT ARGS: Real number — Floating point number to be converted.
                         W — Field width (total number of characters, including
                             sign and decimal point, to be converted).
                         D — Number of characters after decimal point.

OUTPUT ARGS: ARRAY — Filled real ARRAY

COMMON: None

SUBROUTINES USED: PUTC

NOTES: None

EXAMPLE: Convert the real number 123.4 into an ASCII string —
                            with the same format.
                            DIMENSION ARRAY(3)
                                :
                            X = 123.4
                            IWRD = 5
                            ID = 1
                            CALL (X, ARRAY, IWRD, ID)
                                :

                                :

NAME: KAFP

PURPOSE: Converts a string of unpacked ASCII characters
                            (1 character per double word) into a FORTRAN format
                            floating point number.

METHOD: The sign bit is established.   The whole number part
                            is the final sum of each integer times the appropriate
                            power of 10 for its position.   The fractional part is
                            the sum of each fractional digit divided by the appro-
                            priate power of 10.   The whole fractional parts are
                            then merged, normalized, and merged again with
                            the exponent and sign bit.

CALL: CALL: KAFP (ARRAY, COUNT, REAL NUMBER)

RESTRICTIONS: Number range: 0.00001 to 131,071

INPUT ARGS: Array — ASCII character string, 1 character per
                                    double word, left unjustified.
                            Count — Number of characters in array.

OUTPUT ARGS: Real number — FORTRAN format floating point
                            number.

COMMON: None

SUBROUTINES USED: None

EXAMPLE: Convert the four ASCII numbers stored in array
NUMB into a floating point number called "X".

```
        DIMENSION NUMB(4)
        DATA NUMB(1), NUMB(2), NUMB(3), NUMB(4)
    1   /5H1    , 5H2    , 5H3    , 5H. /
            :
            :
        ICNT = 4
        CALL KAFP (NUMB(1), ICNT, X)
            :
            :
```

A.1.4.4 String Manipulation

Five ASCII characters are stored in each word pair in the PDP-9. It takes
52 word pairs to hold the display data to fill the command keyboard display's
256 character positions. Since it is unlikely that display data merged into
a fixed-format display will start and stop on word boundaries, it is necessary
to have the capability to merge ASCII strings regardless of their starting or
ending locations. Three routines were developed for this purpose.

NAME: GETC

PURPOSE: To recreate an ASCII character from the named
position in a packed ASCII double word into the
named location, left unjustified.

METHOD: The character is masked out of the original word,
left-shifted, and stored in the given destination
address.

CALL:	CALL: GETC (source word, position, destination word)
RESTRICTIONS:	None
INPUT ARGS:	Source word — Packed ASCII double word containing character
	Position — Integer, 0-4, defining position in word (left-to-right orientation)
OUTPUT ARGS:	Destination word — Filled word (first word only)
COMMON:	None
SUBROUTINES USED:	None
EXAMPLE:	Load the third character of SWRD into the third word of array ICHAR.

```
DATA SWRD/5HABCDE/
DIMENSION ICHAR(4)
    :
I = 2
    :
CALL GETC (SWRD, I, ICHAR(3))
    :
    :
```

NAME:	PUTC
PURPOSE:	Copies a left-justified character into the named position in an ASCII word pair.
METHOD:	The word pair is masked. The character is positioned. The character is exclusive OR'ed into the word pair.

33

CALL:	CALL: PUTC (double word, position, character)
RESTRICTIONS:	None
INPUT ARGS:	Position — Integer, 0-4, defining destination position of character (left-to-right orientation).
	Character — Left-justified character
OUTPUT ARGS:	Double word — Modified double word
COMMON:	None
SUBROUTINES USED:	None
EXAMPLE:	Merge the two given strings to yield "A=2".

```
DATA DWRD/5HA = ---/, ICHAR/5H2----/
   :
I = 2
   :
CALL PUTC (DWRD, I, ICHAR)
   :
```

NAME:	MOVSTR
PURPOSE:	Copies a string of packed ASCII characters from one array into another — based on row and column positioning (see Notes)
METHOD:	Establishes source array first word and first character position from input array row and column. Establishes destination word and character position. Successively calls GETC and PUTC for total number of characters to be moved, adjusting word locations and character positions as required.

CALL:	CALL: MOVSTR (SOURCE ARRAY, SROW, SCOLUMN, DESTINATION ARRAY, DROW, DCOLUMN, COUNT)
RESTRICTIONS:	Moves no more than 256 characters.
INPUT ARGS:	SOURCE ARRAY — First word of source array (See Notes)
	SROW — Source row
	SCOLUMN— Source column
	DESTINATION ARRAY — First word of destination array
	DROW — Destination row
	DCOLUMN — Destination column
	COUNT — Number of characters to copy
OUTPUT ARGS:	DESTINATION ARRAY — First word of destination array
	DROW — Destination row
	DCOLUMN — Destination column
COMMON:	None
SUBROUTINES USED:	GETC, PUTC
NOTES:	Row and column notation is based on eight rows of 32 characters, column being the position in the row. Row is from 0-7, column from 0-31. This notation is equivalent to PKD cursor position notation. For MOVSTR, arrays are referenced by the first word in the array, with row and column values establishing actual word location.

EXAMPLE: Merge strings to yield "PAGE NO. 2"
                         DATA DA(1), DA(2)/5HPAGE, 5HNO. /, SA(1)/5H12345/
                              :
                         I = 1
                         ISROW = 0
                         ISCOL = 1
                         IDROW = 0
                         IDCOL = 9
                         CALL MOVSTR (SA(1), ISROW, ISCOL, DA(1)
                                      IDROW, IDCOL, I)
                              :
                              :

A. 1. 4. 5 DEC-339 Display Support

Although this CRT device is not used in the Crew/Computer Communications
Study demonstration, it was used in evaluating interface techniques.   The
routine GET339 is used to retrieve the display data from the disk.

NAME: GET339

PURPOSE: Get current CRT display files from the disk.

METHOD: Read the three display files associated with each
                         339 display.

CALL: CALL GET339

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: Labeled common block S1

                          S — Display data for status area or screen

                          A — Display data for action area or screen

                          W — Display data for work area or screen

                          FST — File name of the status area of the screen

                          FAC — File name of the action area of the screen

                          FWK — File name of the work area of the screen

SUBROUTINES USED: None

NOTES: The 339 display screen was partitioned as follows:

Status Area Action Line	6 lines
Work Area	15 lines
Error & Diagnostic Messages	5 lines

EXAMPLE: None

A.1.4.6 Error Handling

Invalid input to the PDP-9 from the PKD is identified and an error message
is displayed on the bottom line of the programmable command keyboard
display. Two routines are used to display the two general class of errors:
(1) transmission errors, and (2) selection errors (i.e., unlabeled key
pressed).

NAME: EROUT1

PURPOSE: Display error messages detected by routine PKIN.

METHOD: Read error code from routine PKIN and display
                          corresponding message on PKD.

CALL: CALL EROUT1

RESTRICTIONS: Only four types of error codes exist—IEC = 1 through 4

INPUT ARGS: IEC — Error code from PKIN

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKCC — Cursor location command routine
                       PKSOUT — String output routine

NOTES: Error codes:
                       0 = None detected
                       1 = Command keyboard display switch value out of
                           range
                       2 = Special function switch value out of range
                       3 = Illegal key type
                       4 = Illegal key type

EXAMPLE: Display error detected by PKIN
                          :
                          :
                   10 CALL PKIN (IST, ITY, IVAL, IIV, IEC)
                       IF (IST. EQ. 0) go to 10
                       IF (IEC. NE. 0) go to 20
                          :
                          :
                   20 CALL EROUT1

NAME: EROUT2

PURPOSE: Display errors on command keyboard display associ-
                       ated with command keyboard selections.

METHOD: Print error message according to code in software
                         status table (last entry found in GENFIL)
                         T = 0. Unlabeled key pressed
                         T = 1 Activity not programmed
                         T = 2 Command valid (0, 1 only meaningful values in
                                                 this program)

CALL: CALL EROUT2(T)

RESTRICTIONS: T = 0 or 1

INPUT ARGS: T = Software status

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKCL — Cursor location command routine
                         PKSOUT — String output routine

NOTES: None

EXAMPLE: None

A. 1. 5 Off-Line Utility Software

All support routines that are not and cannot be called by the application
programs fall in this group.   They are stand-alone routines used for the
following functions:

    A.   Display Generation for DEC-339 and PKD

    B.   Linking Table Generation

    C.   Display Data Check and Hardcopy Printout

    D.   PKD Checkout

NAME: GENPKD

PURPOSE: Generate or edit a PKD display file for storage on
                          disk. This file can obtain up to 256 ASCII characters
                          of data.

METHOD: Load file name and state if it is stored on disk.
                          Specify word start address desired on PKD display.
                          Load display data through TTY keyboard and display
                          on PDK. When data input is complete, type a start
                          address equal to 60. When the request for number of
                          characters is typed, type 255. At this point, the
                          data will be displayed on the command keyboard
                          display.

CALL: None

RESTRICTIONS: If file is stated to exist on disk and it is not found,
                          the monitor will type IOPS 13 (file not found) on the
                          TTY.

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKOUT — Initialize output handler
                          PKPOUT — Page output to PKD display
                          PKCC — Cursor control command routine
                          PKSOUT — String output to PKD display

NOTES: None

EXAMPLE: See User's Guide (Section A. 2).

NAME: GEN339

PURPOSE: Generate relocatable display subpictures for
                    DEC-339 display.

METHOD: FORTRAN-coded 339 commands are used to define
                    up to 3 displays which will be stored on tape for
                    later use.   Display data are defined in data statements.

CALL: None

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PRAMTR — Set display parameters
                    LINE — Draw a line
                    TEXT — Display test string
                    DINIT — Initialize displays
                    SETPT — Locate beam on display surface
                    PLOT — Display subpicture or main picture

NOTES: None

EXAMPLE: See preamble of program listing.

NAME: GENFIL

PURPOSE: Generate linking logic (execution table) for an inter-
                    active program step.   The resulting file will contain:
                    (1) PKD display file name; (2) new file directories

associated with each command key; and (3) the program status for all commands.

METHOD: Read all data from TTY and transfer them to disk when complete.

CALL: None

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: None

NOTES: Output data format:
A. Display file name (A5)
B. File directories (16A5)
C. Software status (16I1)

EXAMPLE: See User's Guide

NAME: CHKPK

PURPOSE: Display requested display file on PKD.

METHOD: A. Read display name from TTY.
B. Read file on disc.
C. Display file on PKD.

CALL: None

RESTRICTIONS: IOPS error 13 results if requested file is not found
                  on disk.

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKOUT — Initialize output handler
                  PKPOUT — Page output to PKD handler

NOTES: None

EXAMPLE: None

NAME: CHK339

PURPOSE: To display relocatable subpictures stored on disk.

METHOD: Subpictures that were defined in GEN339 utilizing
                  DEC-supplied display commands and stored on disk
                  are read back as data and replotted.

CALL:

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PRAMTR — Set display parameters

                                  (set scale, beam intensity, etc...)

                           LINE — Draw a line or move beam

NAME: CHKOUT

PURPOSE: Checkout driving and response routines for PKD.

METHOD: All the cursor control commands are displayed on the command keyboard display and are executed when the corresponding key is pressed. Special function keys are used to turn on or off the LED's above the corresponding keys. Numeric key depressions generate a character at the present cursor position.

CALL: None

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKOUT — Initialize output handler

                           PKPOUT — Page output to PKD display

                           PKIN — Response routine for PKD

                           EROUT2 — Error message routine

                           PKCL — Cursor control command routine

                           PKSOUT — String output to PKD display

                           PKMSL — Special function key light command

                           PKCOUT — Character output routine

NOTES: After CHKOUT is loaded, all keyboards are active and will respond as stated under METHOD.

TEXT — Display text string

DINIT — Initialize display

SETPT — Locate beam on display surface

PLOT — Display subpicture or main picture

REPLOT — Change and display modified subpicture

EXAMPLE: None

NAME: WRTPKD

PURPOSE: Type on TTY the contents of the specified PKD display file that is stored on disk.

METHOD: Type file name at TTY. Read file from disk and type results on TTY console in the same format as the PKD display (32 characters per line).

CALL: None

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: None

NOTES: None

EXAMPLE: None

A. 2 USER'S GUIDE

The objective of this User's Guide is to enable a programmer to understand and use the software developed during this study.

The control commands described in this section of the report are input commands for a TTY console connected to a PDP-9. The following shorthand notation will be used to describe special keyboard commands: (R) for RETURN, (AM) for ALT MODE, and ⌊___⌋ for space.

A. 2. 1 Generating Displays

Displays for the programmable command keyboard are generated with the off-line utility routine, GENPKD. This routine enables one to edit an old or generate a new display file of up to 255 characters. A full 64-character data set is available in a 7-bit ASCII format. Five of these characters are packed in each word pair on the PDP-9, thereby requiring an array of 52 word pairs per display.

The first step in generating a display is to lay it out on a PKD page generation worksheet as shown in Figure A-1. The last character position on the bottom line is reserved for the cursor in full-page displays and is not addressed by routine GENPKD. The first word of the display data array is reserved as a header which is generated in routine PKPOUT.

Routine GENPKD can now be used to define the display data. This is done as follows:

    A.   Load the DEC monitor following standard operating procedures. When loaded, the message "KM59-15 VSA" is typed on the teletype followed by a "$" on the next line.

    B.   If the display files are not to be stored on the same disk at the source programs, the selected disk is identified by typing the following command after the $.

            SA ⌊___⌋ DK# ⌊___⌋ 1 (R)

            Where: # = logical disk number (i. e. , 2, 3, or 4)

LINE 1

LINE 2

LINE 3

LINE 4

LINE 5

LINE 6

LINE 7

LINE 8

*THESE NUMBERS (2, 3, . . . , 52) ARE INDEXES OF THE STARTING POINTS WHICH
THE PAGE GENERATION PROGRAM CAN PROVIDE. TO START AT A POINT ON A LINE NOT
UNDER AN INDEX NUMBER, YOU MUST ACTUALLY START UNDER THE PREVIOUS INDEX,
PROVIDING CHARACTERS AS APPROPRIATE

Figure A-1. PKD Page Generation Worksheet

C. If a disk assignment was made as above, another "$" will be typed
by the computer. The system loader is now read into core by
typing the following command after the $.

$GLOAD (R)

D. After the computer responds by typing a ">," the display genera-
ting program is loaded by typing:

> ← GENPKD, PDS (AM)

E. When GENPKD is loaded, the following message is typed:

FILE NAME IS = (A5)

The file name specified on the page generation worksheet is now
typed, followed by pressing the carriage return. This file name
can be up to five alphanumeric characters long, although the first
character must be alpha.

F. The system responds to this input with the following message:

IF DISPLAY IS STORED ON DISK, TYPE 1,

IF NOT, TYPE 0.

Type "1" for an existing file or "0" to generate a new display file
and then press the carriage return.

47

G. The computer will then request the start position in the display array by typing the following message:

LOAD ASCII DATA

LINE START SUBSCRIPT =

If this display is new, type in the first addressed location on the PKD page generation worksheet defined in Figure A-1; however, if an old display file is being edited, it is only necessary to type in the address of the word to be modified. It should be noted that all edited input data start at a word boundary corresponding to the subscript that is typed. However, when the first desired location is not on a word boundary, characters must be filled in starting with the previous word boundary. After the subscript is typed, press carriage return.

H. A position reference string is then typed by the computer identifying each of the 30-character input positions followed by the TTY executing a carriage return. If a subscript value of 10 had been typed, the TTY would have responded with:

10 --- 11 --- 12 --- 13 --- 14 --- (R)

I. The display data are then typed under the corresponding character position in the string. After the 30 positions have been loaded, the carriage return is pressed.

J. The computer then responds with the same message as step G and the three previous instructions are repeated until all the display has been defined. The user leaves this loop by typing a subscript value of 60, followed by a carriage return.

K. The computer will then ask for the number of characters in the display. Type 255 for a full page or the number of characters for string data. Follow this input with a carriage return.

L. The computer will now display the data on the PKD and the following message will be typed:

TYPE 1 TO LOAD DISK OR 2 TO CHANGE DISPLAY

After typing in the desired value, press carriage return.

M. If 1 was typed in the prior step, the message in step 5 is typed, indicating the display is defined and stored in the disk. The computer is now ready to accept a new file name. Proceed as described in

steps 5 through 12. If, however, a 0 was typed in step 12, a return to step 7 is executed and a request will be made for the subscript of the line to be corrected.

A.2.2 Generating Execution Tables

The execution table is the media for linking displays and defining the software status of programmable command keyboard keys. These tables are generated by routine GENFIL. In its present form, these data are confined to actions on the programmable command keyboard. The execution table, as shown below, is 33 words long.

Present PKD Display File name	1 word
Execution table file names for all active keys	16 words
Software status of all keys	16 words

Execution table data should be laid out on a data sheet, as shown in Figure A-2.

Routine GENFIL can now be used to define the execution tables. This is done as follows:

A. Load the DEC monitor per standard operating procedures. When loaded, the message "KMS9-15 VSA" is typed on the teletype followed by a "$" on the next line.

B. If the display files are not to be stored on the same disk as the source programs, the selected disk is identified by typing the following command after the "$":

$A⌐⌐ DK#⌐⌐1 (R)

Where: # = logical disk number (i.e., ... 2, 3, or 4)

C. If a disk assignment was made as above, another "$" will be typed by the computer. The system loader now is read into core by typing the following command after the "$":

$GLOAD (R)

D. After the computer responds by typing a ">" the display generating program is loaded by typing:

> ←GENFIL (AM)

FILE NAME =

    A. DISPLAY FILE NAME =

    B. FILE DIRECTORY CMD/KEY

KEY NO.	DIRECTORY NAME	KEY NO.	DIRECTORY NAME
1		9	
2		10	
3		11	
4		12	
5		13	
6		14	
7		15	
8		16	

    C. SOFTWARE STATUS

        0 — UNLABELED KEY (DEFAULT)

        1 — ACTIVITY NOT PROGRAMMED

        2 — KEY VALID

KEY NO.	STATUS	KEY NO.	STATUS
1		9	
2		10	
3		11	
4		12	
5		13	
6		14	
7		15	
8		16	

Figure A-2. Genfil Code Sheet

E. When GENFIL is loaded the following message is typed:

> PROGRAM: GENFIL
>
> FILE NAME IS = (A5)

The file name specified on the GENFIL code sheet is now typed, followed by pressing the carriage return. This file name can be up to five alphanumeric characters long, although the first character must be alpha.

F. The computer then requests the name of the display file that is active when this execution table is used by typing:

> PKD DISPLAY FILE = (A5)

The five alphanumeric characters representing this file name (Item A — Figure A3) are typed, followed by pressing the carriage return.

G. The computer responds by requesting the execution file names of all "active" programmable command keys. To do this the computer first asks:

> FILE DIRECTORY CMD/KEY
>
> CMD-KEY NO. =   (R)

It is only necessary to load the "active" key numbers. If no keys are active or all the keys have been identified, type 17 (R). Active keys will cause the computer to ask for the file names by typing the following message:

> FILE NAME =

Load this information and press the RETURN key. A request will be made for a new command key as above, and the cycle repeats until a value of 17 is loaded.

H. A request will now be made by the computer to define the software status of the keys. A message appears as follows:

> SOFTWARD STATUS
>
> 0 = UNLABELED KEY
>
> 1 = CMD. NOT PROGRAMMED
>
> 2 = CMD. GOOD
>
> CMD-KEY NO. =

It is only necessary to identify those keys which are labeled in the display that this execution table supports as the default condition is zero. Type in the key number, followed by a carriage return. The computer will now request the software status by typing the following message:

SOFTWARE STATUS =

Type in the value followed by a carriage return. A new request will then be made by the computer for another command key value. Repeat this cycle until all keys have been identified, at which time a value of 17 is typed, followed by a carriage return.

I.   The computer types the following message to indicate that the execution table is complete:

FILE IS COMPLETE
PROGRAM: GENFIL
FILE NAME =   (A5)

If the user wishes to define a new execution table for another display, he may do so at this time; otherwise, he can terminate the program by pressing CTRL P.

A. 2. 3   The Application Executive

The operation of the LM/CSM rendezvous application executive, MAIN, is depicted in Figure A-3. It should be noted that it is the user's responsibility to poll the input handler for PKD responses. This is accomplished by calling routine PKIN. This routine will tell the user if there was a valid input command from the PKD and if so, it will decode it. Either invalid input or no input results in continued polling.

Valid data are tested to determine which keyboard generated the interrupt. If it was a special function key, the routine MODKEY is called with the key number defined in the argument. This routine performs the programmed function and returns to the polling routine.

Numeric keyboard input is only valid when routine NUMIN is called. This routine takes care of blinking the cursor at a predefined location on the

52

```
                    ┌─────────┐
                    │  START  │
                    └────┬────┘
                         │
                 ┌───────▼───────┐
                 │  INITIALIZE   │
                 │   SOFTWARE    │
                 └───────┬───────┘
                         │
                 ┌───────▼───────┐
                 │ READ DISPLAY  │
                 │   DATA AND    │
                 │  EXECUTION    │
                 │    TABLE      │
                 └───────┬───────┘
                         │
                 ┌───────▼───────┐
                 │   DISPLAY     │
                 │    DATA       │
                 │   ON PKD      │
                 └───────┬───────┘
```

A	

- READ PKD RESPONSE
- IS THERE NEW INPUT — NO / YES
- IS INPUT VALID — NO / YES
- CALL NUMERIC KEY ROUTINE
- IS KEYBOARD ACTIVE — YES / NO
- CALL ERROR ROUTINE
- NUMERIC KEYBOARD
- PROGRAMMABLE COMMAND KEYBOARD
- PROCESS INPUT
- CALL SPECIAL FUNCTION KEY ROUTINE
- IS KEY ACTIVE
- CALL ERROR ROUTINE → **A**
- NEW DISPLAY REQUIRED — YES / NO → **A**
- IS DISPLAY "CHECKOFF LIST" — YES / NO → **1**
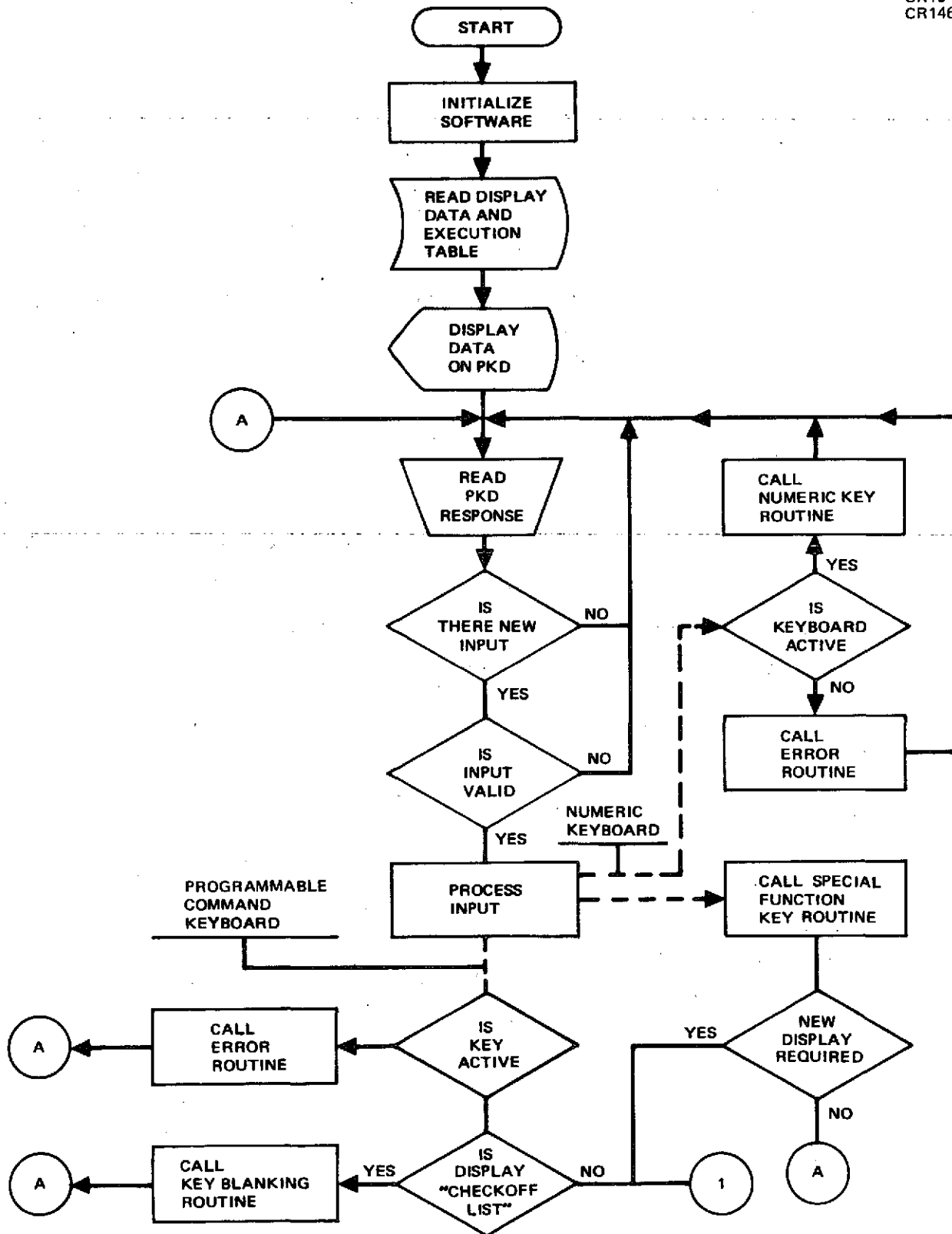- CALL KEY BLANKING ROUTINE → **A**

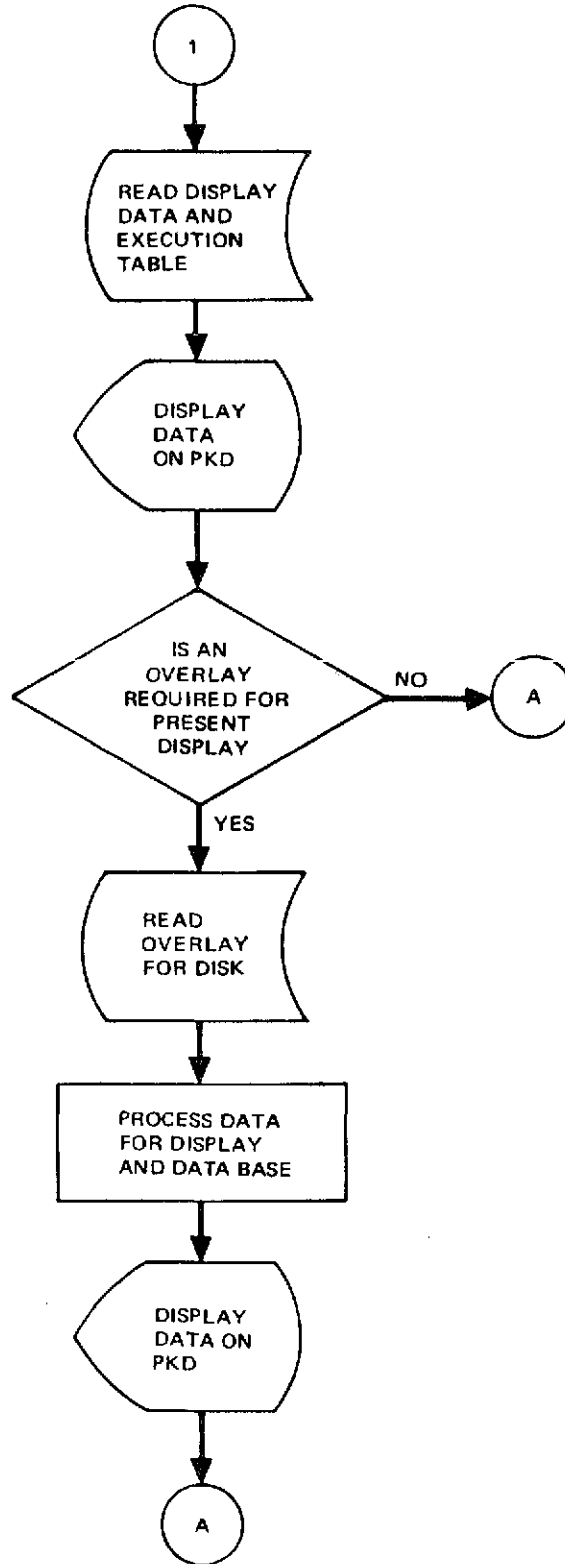Figure A-3 (Page 1 of 2). Application Executive Flow Chart

Figure A-3 (Page 2 of 2). Application Executive Flow Chart

programmable command keyboard and displaying characters as they are
loaded. The ASCII input characters are also packed into word pairs, left-
justified. Numeric key input at any other time will result in NUMKEY being
called, which is equivalent to calling an error routine. The next revision of
the application software will combine NUMIN and NUMKEY with a flag to
test when the numeric keyboard is active.

Programmable command keyboard input generally will result in changing
the present display. Some displays require updating and displaying data
base parameters. The required in-line code for updating these displays is
partitioned into core overlays. A table exists in the resident program
MAIN, defining the file names of all the displays requiring core overlay
support code. When a programmable command keyboard input is received
from the PKD, it is tested to see if the new display requires an overlay to
be loaded. If the display file name matches any of those in the list, the
required overlay is called. Since a single overlay might support three or
four displays, it is necessary to perform a test at the beginning of the
overlay to determine which display support code to branch to.

A. 2. 4 Running the Program on the PDP-9
The "Chain and Execute" utility routines are used for the LM/CSM
rendezvous application. The resident code during execution would include the
following routines: MAIN, EROUT1, EROUT2, NUMIN, NUMKEY,
MODKEY, GETPKD, PDS, BLINKY, KFPA, KAFP, and TIMCON. The
rendezvous calculations are performed in the second overlay by RENCAL,
with the results transferred to the first via unlabeled COMMON. All of
these routines are stored on Logical Disk 1. Display data and execution
tables are stored on Logical Disk 3.

APPENDIX B
PROGRAMMABLE KEYBOARD AND DISPLAY

Appendix B
PROGRAMMABLE KEYBOARD AND DISPLAY

The programmable keyboard and display (PKD) was developed by McDonnell Douglas to implement the man-machine interface required in this study. The PKD is a compact, portable command, control, and display terminal employing state-of-the-art hardware. It provides sufficient display and control to support complex communications between operator and computer. A photograph of the PKD is shown in Figure B-1. The PKD is designed for direct connection to a computer, and for remote operation when an acoustic coupler and voice grade telephone lines are used for the computer link (see Figure B-2).

The PKD is adaptable to command and control systems for spacecraft, aircraft, ground checkout and launch, and defense systems, as well as mass transportation systems and communications networks. The ability to vary the display from the computer and to receive an operator's reply in the form of a "touch" input makes the unit flexible and easy to use.

The salient features of the PKD are a 256-character alphanumeric display (8-row, 32-column matrix), a numeric keyboard, and 12 special-function switches. In addition, the display panel includes a transparent overlay of conductors to form a 4 x 4 matrix of touch switches, each switch area containing two rows of eight characters (see Figure B-3). The display and controls of the PKD allow for all the major functions of a CRT terminal except graphics, although the PKD could be used as a controller for graphics displays. The PKD block diagram appears Figure B-4.
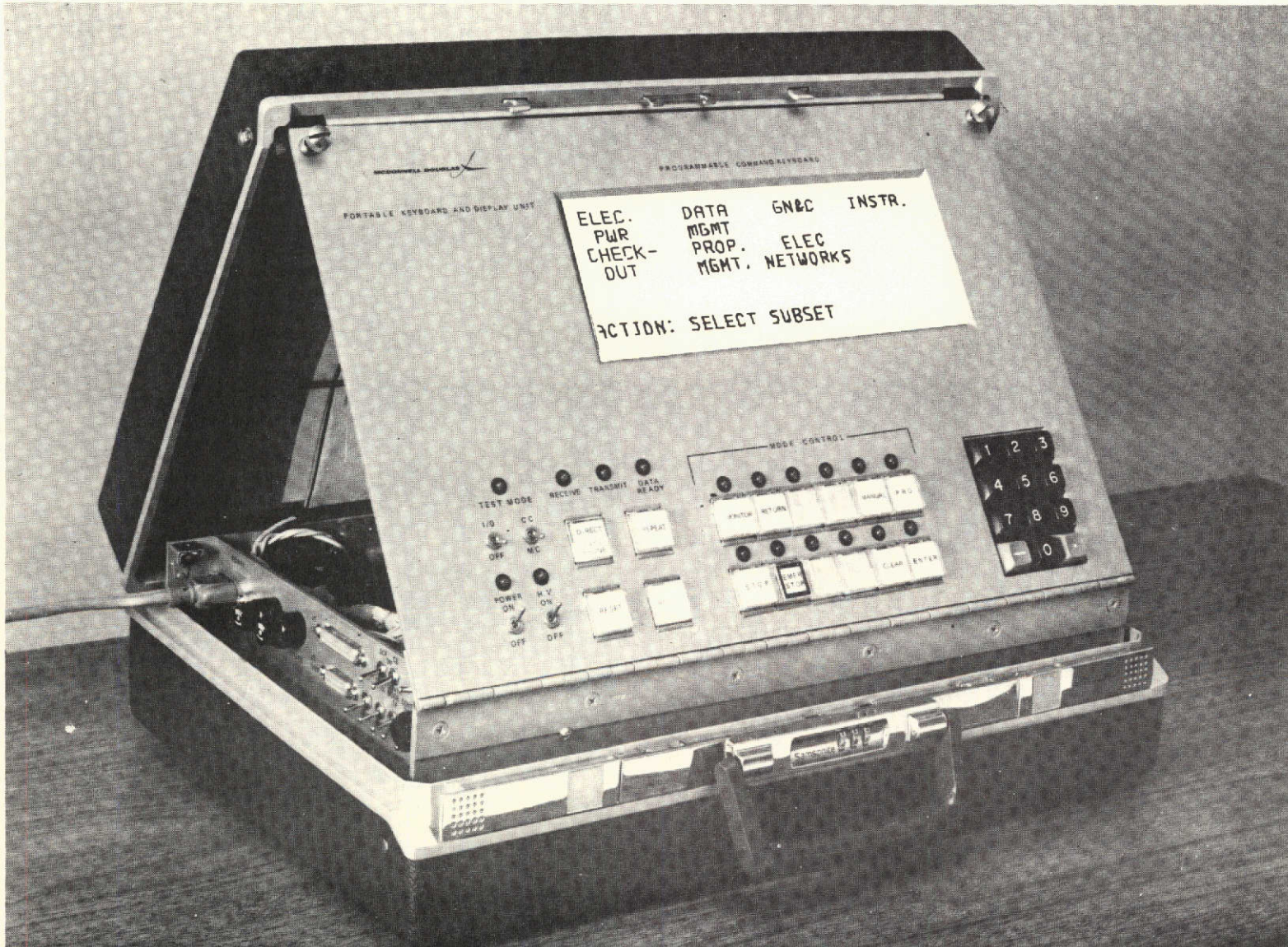
Figure B-1. Programmable Command Keyboard

ACOUSTIC
COUPLER

DIRECT CONNECTION

ACOUSTIC
COUPLER

EIA LEVEL
CONTROL

CONTROL
LOGIC

I/O INTERFACE
AND
CONTROL

MASS
STORAGE
DEVICES

I/O
CHANNEL

CPU

COMPUTER

Figure B-2. PKD/Computer Data Transmission

COLUMN:

0 1 2 3 4 5 6 7 8   15 16   23 24   31

ROW:
0

1

2

3

4

5

6   A B C D E F G H

7   1 2 3 4 5 6 7 8

TOUCH SWITCH
BOUNDARIES

Figure B-3. Transparent Overlay of Conductors

Figure B-4. Block Diagram-Portable Keyboard and Display Unit (PKD)

The 256-character display is a Burroughs SELF-SCAN panel display. The display consists of a matrix of gas-filled cells spaced at 0.040-inch centers. The character format is a 5 x 7 dot matrix. The panel provides a character siz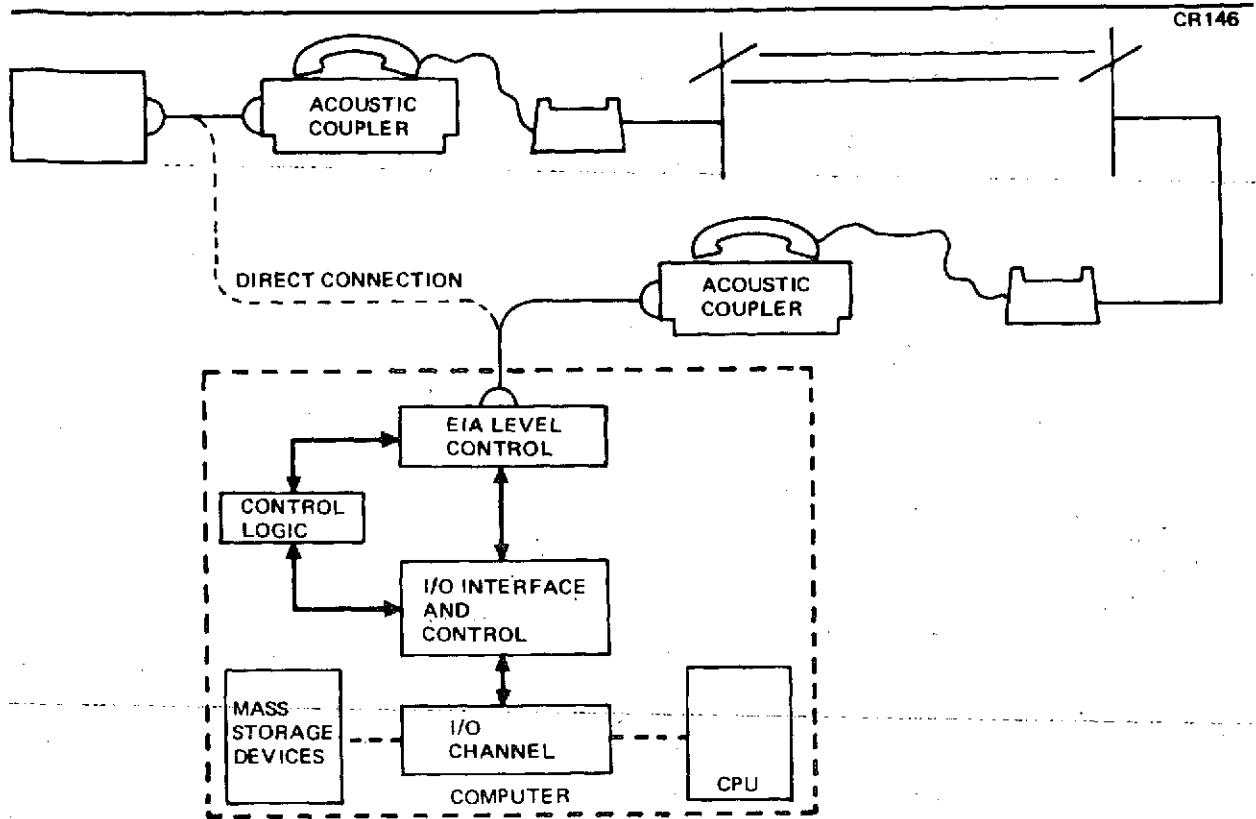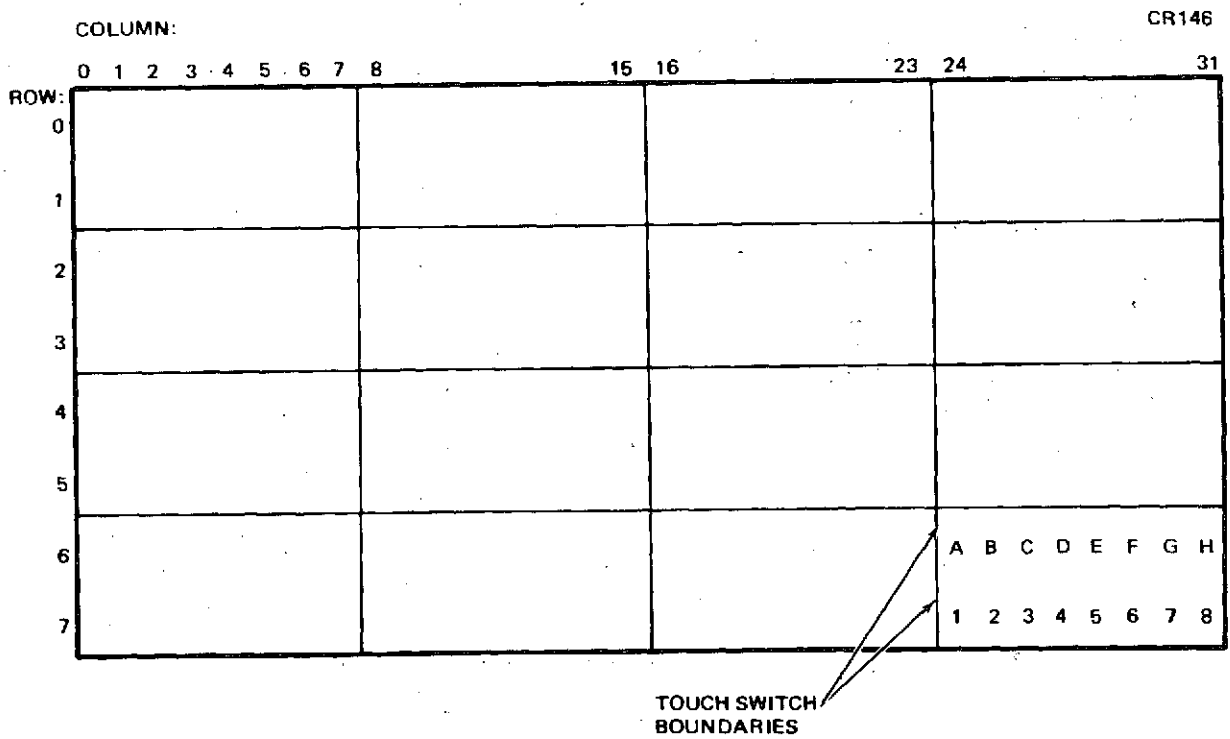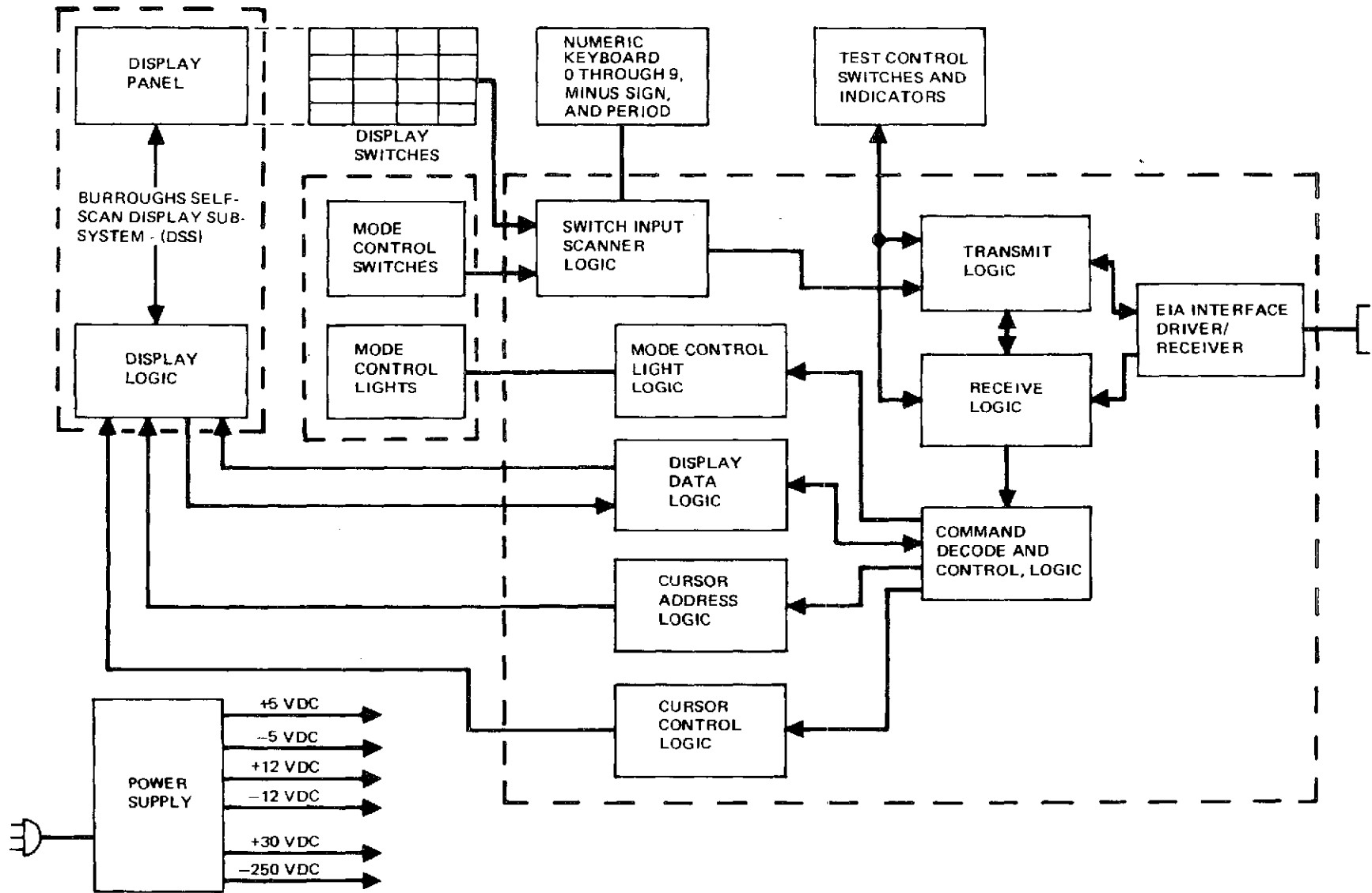e of 0.18-inch wide by 0.28-inch high. The displayed characters are retained in a 6 x 256 memory array for self-refresh. The flat panel accommodates a 100-degree viewing angle (minimum).

The touch switch overlay added by McDonnell Douglas enhances the human interface through reliance on the natural coordinated motion of the hand and eye. By means of appropriate software, the switch areas can be filled with commands, path descriptions, or program controls whose selection will be recognizable internally, thereby controlling the computer system activity.

The PKD also has built-in self-test capability. Without the aid of a computer, a self-test mode can be established with control switches. In self-test, the inputs from the operator are transmitted through the PKD output circuits and then looped back through the input. An internal 10-kHz clock provides timing. The input word is transmitted through the receiver circuit to either the display panel or the panel indicator lights as determined by the code received. The self-test capability is complete in that all receivable information can be generated by the operator, and all output words from the PKD can be verified for accuracy on the face of the unit.

The eight bit words received by the PKD are of four types:
  A. Display data — Modified six-bit ASCII code (see Figure B-5) representing the character to be displayed at the current cursor location.
  B. Special function light commands — Turn on/turn off command to the selected light.
  C. Cursor position command — Position cursor at indicated row (one of eight) and indicated column (one of 32).
  D. Cursor control command — Twelve commands for cursor spacing, relocating, and turn on/off.

Input/output data specifications appear in Figure B-6.

8	7	6	5	4	3	2	1	CHARACTER		8	7	6	5	4	3	2	1	CHARACTER
0	1	0	0	0	0	0	0	@		0	0	1	0	0	0	0	0	(SPACE)
0	1	0	0	0	0	0	1	A		0	0	1	0	0	0	0	1	!
0	1	0	0	0	0	1	0	B		0	0	1	0	0	0	1	0	"
0	1	0	0	0	0	1	1	C		0	0	1	0	0	0	1	1	#
0	1	0	0	0	1	0	0	D		0	0	1	0	0	1	0	0	$
0	1	0	0	0	1	0	1	E		0	0	1	0	0	1	0	1	%
0	1	0	0	0	1	1	0	F		0	0	1	0	0	1	1	0	&
0	1	0	0	0	1	1	1	G		0	0	1	0	0	1	1	1	'
0	1	0	0	1	0	0	0	H		0	0	1	0	1	0	0	0	(
0	1	0	0	1	0	0	1	I		0	0	1	0	1	0	0	1	)
0	1	0	0	1	0	1	0	J		0	0	1	0	1	0	1	0	*
0	1	0	0	1	0	1	1	K		0	0	1	0	1	0	1	1	+
0	1	0	0	1	1	0	0	L		0	0	1	0	1	1	0	0	,
0	1	0	0	1	1	0	1	M		0	0	1	0	1	1	0	1	-
0	1	0	0	1	1	1	0	N		0	0	1	0	1	1	1	0	.
0	1	0	0	1	1	1	1	O		0	0	1	0	1	1	1	1	/
0	1	0	1	0	0	0	0	P		0	0	1	1	0	0	0	0	0
0	1	0	1	0	0	0	1	Q		0	0	1	1	0	0	0	1	1
0	1	0	1	0	0	1	0	R		0	0	1	1	0	0	1	0	2
0	1	0	1	0	0	1	1	S		0	0	1	1	0	0	1	1	3
0	1	0	1	0	1	0	0	T		0	0	1	1	0	1	0	0	4
0	1	0	1	0	1	0	1	U		0	0	1	1	0	1	0	1	5
0	1	0	1	0	1	1	0	V		0	0	1	1	0	1	1	0	6
0	1	0	1	0	1	1	1	W		0	0	1	1	0	1	1	1	7
0	1	0	1	1	0	0	0	X		0	0	1	1	1	0	0	0	8
0	1	0	1	1	0	0	1	Y		0	0	1	1	1	0	0	1	9
0	1	0	1	1	0	1	0	Z		0	0	1	1	1	0	1	0	:
0	1	0	1	1	0	1	1	[		0	0	1	1	1	0	1	1	;
0	1	0	1	1	1	0	0	~		0	0	1	1	1	1	0	0	<
0	1	0	1	1	1	0	1	]		0	0	1	1	1	1	0	1	=
0	1	0	1	1	1	1	0	{		0	0	1	1	1	1	1	0	>
0	1	0	1	1	1	1	1	}		0	0	1	1	1	1	1	1	?

Figure B-5. Modified ASCII Codes

I. INPUT/OUTPUT DATA DEFINITION

A. ELECTRICAL INTERFACE

ELECTRICAL INTERFACE CHARACTERISTICS ARE COMPATIBLE WITH EIA RS—232—C

B. DATA RATE

1. DATA PHONE MODE

HALF DUPLEX WITH APPROPRIATE CONTROL SIGNALS

1,200 BAUD TRANSMIT OR RECEIVE

5 BAUD SUPERVISORY REVERSE CHANNEL

2. DIRECT MODE

FULL DUPLEX, NO CONTROL SIGNALS

10K BAUD RECEIVE

1,200 BAUD TRANSMIT

C. DATA FORMAT

1 START BIT

8 DATA BITS (NO PARITY BIT)

1 STOP BIT
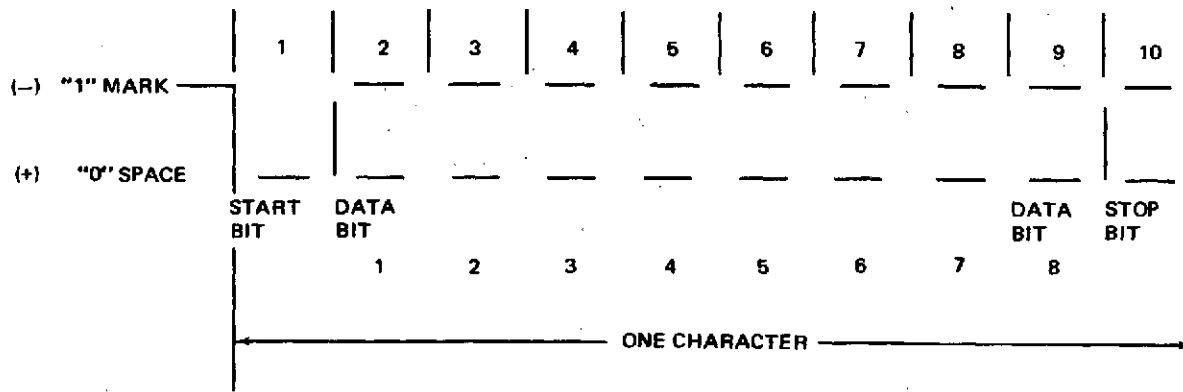
Figure B-6. Portable Keyboard and Display Unit (Page 1 of 2)

D. DATA CODE

DISPLAY SWITCHES

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

2. OUTPUT FROM PKD

DATA BITS								FUNCTION	SPARE
8	7	6	5	4	3	2	1		
0	0	0	0	S	S	S	S	DISPLAY SWITCHES 0–15	– –
0	0	0	1	M	M	M	M	MODE CONTROL SWITCHES 0–11	12–15
0	0	1	0	K	K	K	K	KEYBOARD 0 THROUGH 9, MINUS SIGN, AND PERIOD	12–15
1	X	X	X	X	X	X	X	RESERVED FOR ASC11 KEYBOARD	

3. INPUT TO PKD

DATA BITS								FUNCTION
8	7	6	5	4	3	2	1	
0	0	0	0	M	M	M	M	MODE CONTROL LIGHTS OFF 0–11
0	0	0	1	M	M	M	M	MODE CONTROL LIGHTS ON 0–11
0	0	1	D	D	D	D	D	DISPLAY DATA
0	1	0	D	D	D	D	D	MODIFIED ASC11 (SEE FIGURE B–5) ACCEPTS BITS 1–6 ONLY
0	1	1	0	C	R	R	R	CURSOR ADDRESS LEAST SIGNIFICANT COLUMN BIT + ROW
0	1	1	1	C	C	C	C	CURSOR ADDRESS 4 MOST SIGNIFICANT COLUMN BITS
1	0	0	0	C	C	C	C	CURSOR CONTROL

0	CLEAR
1	HOME
2	VISUAL CURSOR ON
3	VISUAL CURSOR OFF
4	CURSOR BLINK ON
5	CURSOR BLINK OFF
6	CARRIAGE RETURN LINE FEED
7	(SPARE)
8	CURSOR RIGHT
9	CURSOR LEFT
10	CURSOR UP
11	CURSOR DOWN

Figure B-6. Portable Keyboard and Display Unit (Page 2 of 2)

APPENDIX C
PROGRAM DESCRIPTIONS AND USER'S
GUIDE — PART III, PHASE A

Appendix C
PROGRAM DESCRIPTIONS AND USER'S GUIDE
PART III, PHASE A

Section C. 1 describes the software routines developed during the crew/ computer experiment control study. This section contains the standardized program description sheet developed in Appendix A. Section C. 2 describes how these routines are used in the experiment control application.

C. 1 PROGRAM DESCRIPTIONS

The programs described in this section were developed for Part III, Phase A. All previously developed programs appear in Appendix A. See Section A. 1 for programs listed under "Subroutines Used."

The programs which follow are in two categories, application software and off-line utility software, as defined in Section A. 1. Programs KOPE, FILNM, and LENK are the application programs added for the experiment. KOPE is a general-purpose routine for execution control of any application and as such is application-independent. FILNM is a support routine for KOPE and is also application-independent. The LENK routine is exclusively an experiment control routine. The utility program, FGEN, creates display files for the KOPE program so it too is application-independent. The PKD keyboard input routine, KYBIN, is an updated version of the NUMIN routine described in Section A. 1 and is essentially a replacement for that routine.

NAME: KOPE

PURPOSE: Controls execution of application display commands
                             and subroutines. This program is designed to
                             respond to the display data, as defined in the User's
                             Guide FGEN section.

METHOD: Initializes the PKD for transmission and sets local
                             variables to initial values. The program then enters
                             a loop in which the current frame is displayed, after
                             which operator inputs are requested. The response
                             to an input is determined by both the type and value
                             of the PKD input and the values in the display com-
                             mand and pointer tables.

CALL: None

RESTRICTIONS: None

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: OLCOM, FRAME

SUBROUTINES USED: KYBIN
                             FILNM
                             PKOUT
                             PKSOUT
                             PKIN
                             PKCC
                             PKCL
                             KFPA
                             MOVSTR
                             LENK

NOTES: See User's Guide

NAME: FILNM

PURPOSE: Creates a PDP-9 ASCII binary file name from an
                            integer identifier.

METHOD: The integer number input is converted to a multiple
                            of 10. The second, third, fourth, and fifth charac-
                            ters in the name are the ASCII equivalents of the
                            integer (with the fifth character always zero). The
                            characters are inserted in the file name from right
                            to left and the first character set to "F." The file
                            type "BIN" is appended to complete the name.

CALL: CALL FILNM (INTEG, ASCII)

RESTRICTIONS: I ≥ 0

INPUT ARGS: INTEG - integer file number
                            ASCII - real variable or double word

OUTPUT ARGS: ASCII - double word containing file name
                                      in PDP-9 format

COMMON: None

SUBROUTINES USED: MOVSTR - character string mover

NOTES: None

NAME: LENK

PURPOSE: Performs the special functions of experiment control
                         which are beyond the scope of the executive.

METHOD: Branches or program pointers passed from KOPE.
                         Uses local array SW to check for previous activity
                         on a frame. Adjusts values in common to indicate
                         next action of executive after return.

CALL: CALL LENK (IARG1, IARG2)

RESTRICTIONS: None

INPUT ARGS: IARG1   -   value of second entry in program pointer
                                     array
                         IARG2   -   value of third entry in program pointer
                                     array

OUTPUT ARGS: None

COMMON: OLCOM, FRAME (same as KOPE)

SUBROUTINES USED: None

NOTES: None

NAME: FGEN

PURPOSE: Generates or permits editing of PKD display files
                            on disk.

METHOD: Uses PKD input and teletype input to establish con-
                            tents of the display and display pointers. Displays a
                            control frame to permit input of existing or new
                            frame. Accepts commands from the PKD touch
                            switches and mode control switches. Display data
                            are entered from the teletype.

CALL: None

RESTRICTIONS: PKD must be on-line.

INPUT ARGS: None

OUTPUT ARGS: None

COMMON: None

SUBROUTINES USED: PKOUT
                            PKSOUT
                            PKCC
                            PKIN
                            KFPA
                            MOVSTR

NOTES: See User's Guide

NAME:	KYBIN
PURPOSE:	To read in a number from the PKD keyboard, displaying and editing as required, and then return the character string and floating point number after the enter mode switch is received.
CALL:	KYBIN (IR, IC, NC, FPN, ASC, TYPE, VAL)
INPUT ARGS:	IR - Display row starting position for input.
	IC - Display column starting position for input.
	NC - Maximum number of characters to be input, including sign (mandatory) and decimal point characters. NC must be 2 or more and no larger than 10.
OUTPUT ARGS:	FPN - Floating point value of ASCII string or 0.0 otherwise.
	ASCII character string received from PKD Keyboard. Five or 10 characters are returned depending on NC. Trailing blanks are added if NC is not a multiple of 5.
	TYPE - Set to 0 for normal termination of input (by ENTER switch). Otherwise, equals last type input by PKIN.
	VAL - Equals last value received from PKIN.
RESTRICTIONS:	A. ASC must be two double words long if NC is more than 5, one double word otherwise.
	B. $0 \leq IR \leq 7$ C. $0 \leq IC \leq 31$ D. $2 \leq NC \leq 10$
METHOD:	The input parameters are checked for legal values. Program parameters are initialized. ASC is blanked. The cursor is positioned for input. The first character is reserved for the sign. A loop is entered for taking PKD inputs. Exit from the loop is by mode 1

70

input or legal mode 2 input. During input, mode 2
switches CLEAR, ENTER, and BACKSPACE
are enabled to perform their respective functions.

C.2 USER'S GUIDE

This User's Guide was prepared to help a programmer understand and use the
software developed in this study.

The control commands described in this section of the report are input com-
mands for a teletype console connected to a PDP-9. The following shorthand
notation will be used to describe special keyboard commands: (CR) for
RETURN, (AM) for ALT MODE, and /_/ for SPACE.

C.2.1 Generating Frames

Displays for the PKD are generated with the off-line utility routine, FGEN.
This routine enables one to edit or generate a new display frame of up to
255 characters and to assign values to its supporting tables.

The first step in generating a frame is to lay it out on a PKD application
definition worksheet, as shown in Figure C-1. The first word of the display
array is reserved as a header which is generated in the WRITE routine.

Routine FGEN can now be used to define the display data. This is done as
follows:

A.   Load the PDP-9 monitor according to standard operating
     procedures. When loaded, the message "KM59-15 V5A" is
     typed on the teletype, followed by a "$" on the next line.

B.   Assign the disks by typing the following command after the $.

     A/ _/ DKA3/ _/3/ DKA1/ _/1, -4 (CR)

C.   Another $ will be typed by the computer. The execute package is
     read into core by typing the following command after the $.

     E/ _/GEN/ _/ (CR)

71

FRAME NUMBER (15) _____
DISPLAY DATA

|1| |2| |3| |4| |5| |6| |7| |8| |9| |10| (blank grid, 10 rows)

COMMAND POINTERS (I4)

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

FUNCTION POINTERS (I4)

MON 1	UP 2	CAT 3	4
MAN 5	PRO 6	STOP 7	EMER 8
RET 9	DATA 10		
CLEAR	ENTER		

SCRATCH (F10.3)

KBD 1	FRAME 2	C OR F 3	4
5	6	7	8
9	10	11	12
13	14	15	16

PROGRAM (3I4)

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figure C-1. PKD Application Definition Worksheet

D. The computer responds by typing:

"EXECUTE V4A" on the teletype
and displays frame generation tasks on the PKD. (See Figure C-2.).

E. To build a frame press "FETCH =" on the PKD, enter frame number from the numeric keyboard. If it is a new frame, a frame not programmed, the message "function not programmed" is displayed on the PKD.

F. Press "FRAME =" on PKD. Enter frame number from numeric keyboard.

G. The frame data are divided into five programmable sections: display, command pointers, function pointers, scratch, and program (see Figure C-1). To program the display table, press "DISPLAY" on the PKD. To enter data for odd-numbered lines, press the left-most sector of the line. To enter data for even-numbered lines, press the right-most sector of the line. After pressing the sector, the teletype is enabled for data entry. Enter characters from teletype. When the entry is complete, press (CR) on teletype. The text will be displayed on the PKD as well as being entered in the display table. To enter text for lines 9 and 10, press "PROCEED." The PKD display is cleared. Line 9 will be programmed on line 1 of the PKD and line 10 will be programmed on line 2 of the PKD. When lines 9 and 10 are complete, press "PROCEED" to return to task selection.

H. To program the command pointer table, press "COMMAND" on the PKD. Contents of the table are displayed on the PKD. To enter a number into a pointer location, press the "DATA" function key which enables the numeric keyboard. Enter a number. The number is entered into a command pointer location by pressing the sector on the PKD which corresponds to the pointer location. To program command pointers 17 through 20, press "PROCEED." The PKD

73

F	E	T	C	H	=										F	R	A	M	E	=										
C	L	E	A	R	'			P	R	I	N	T			D	E		E	T	E				F	I	L	E			
D	I	S	P	L	A	Y	P	R	I	N	T		A	L	L															
C	O	M	M	A	N	D	F	U	N	C	T	I	O	N	S	C	R	A	T	C	H	P	R	O	G	R	A	M		

Figure C-2. FGEN Task Display

display will be cleared. Sectors 1 through 4 on the PKD correspond to command pointer locations 17 through 20. When data entry is complete, press "PROCEED" to return to task selection.

I.   To program function pointers, press "FUNCTION" on the PKD. Function pointers 1 through 10 are displayed on the PKD. Press the "DATA" function key to enable the numeric keyboard. Enter number from numeric keyboard. To enter a number into a function pointer location, press the sector on the PKD which corresponds to that location, Press "PROCEED" to return the task selection.

J.   To program the scratch table, press "SCRATCH" on the PKD display. Scratch locations 1 through 16 are displayed on the PKD. Press the "DATA" function key to enable the numeric keyboard. Enter the number from the numeric keyboard. PKD sectors 1 through 16 correspond to scratch locations 1 through 16. Press the desired sector to enter the number into the scratch table. Press "PROCEED" to return to task selection.

K. To enter data into the program table, press "PROGRAM" on the PKD display. Program locations 1 through 16 will be displayed on the PKD. Press the "DATA" function key to enable the numeric keyboard. Enter three numbers from the numeric keyboard. The numbers correspond to the three parts of the PKD program instruction word; INSTR, IARG1, and IARG2. To enter the numbers into the program table, press the sector on the PKD which corresponds to the desired program table location. Press "PROCEED" to return to task selections.

L. To write a new or edited PKD frame, press "FILE" on the PKD display.

M. To obtain a hard copy of the current frame in the core, press "PRINT" on the PKD display. To obtain a hard copy of all the PKD frames, press "PRINT ALL" on the PKD display.

N. To edit an existing frame, perform steps A through K with the exception of step F.

C.2.2 The PKD Application Executive (KOPE)

The operation of the PKD executive is indicated in Figure C-3. Upon program initiation, PKD frame 1 is read into the core. The instruction contained in program table location 1 is executed. Successive program instructions, shown in Table C-1, are executed until an instruction with an operation code of 16 is encountered. Control is then passed to the PKD input handler, PKIN, which polls the PKD until a valid input is received.

Valid data are tested to determine which of the PKD switches generated the interrupt. The switch generating the interrupt points to its respective table and the number returned from PKIN points to a location in the table. If the contents of the table location are negative, its absolute value points to a program table location which contains the next PKD instruction to be executed. Execution of PKD instruction 16 will return control to PKIN. If the contents of the calling function's location are greater then zero, the frame corresponding to the number contained in the table location is moved into the core. The instruction contained in program location 1 is executed. If the contents of the instruction executed are 000, control returns to PKIN; otherwise, the next instruction is executed.
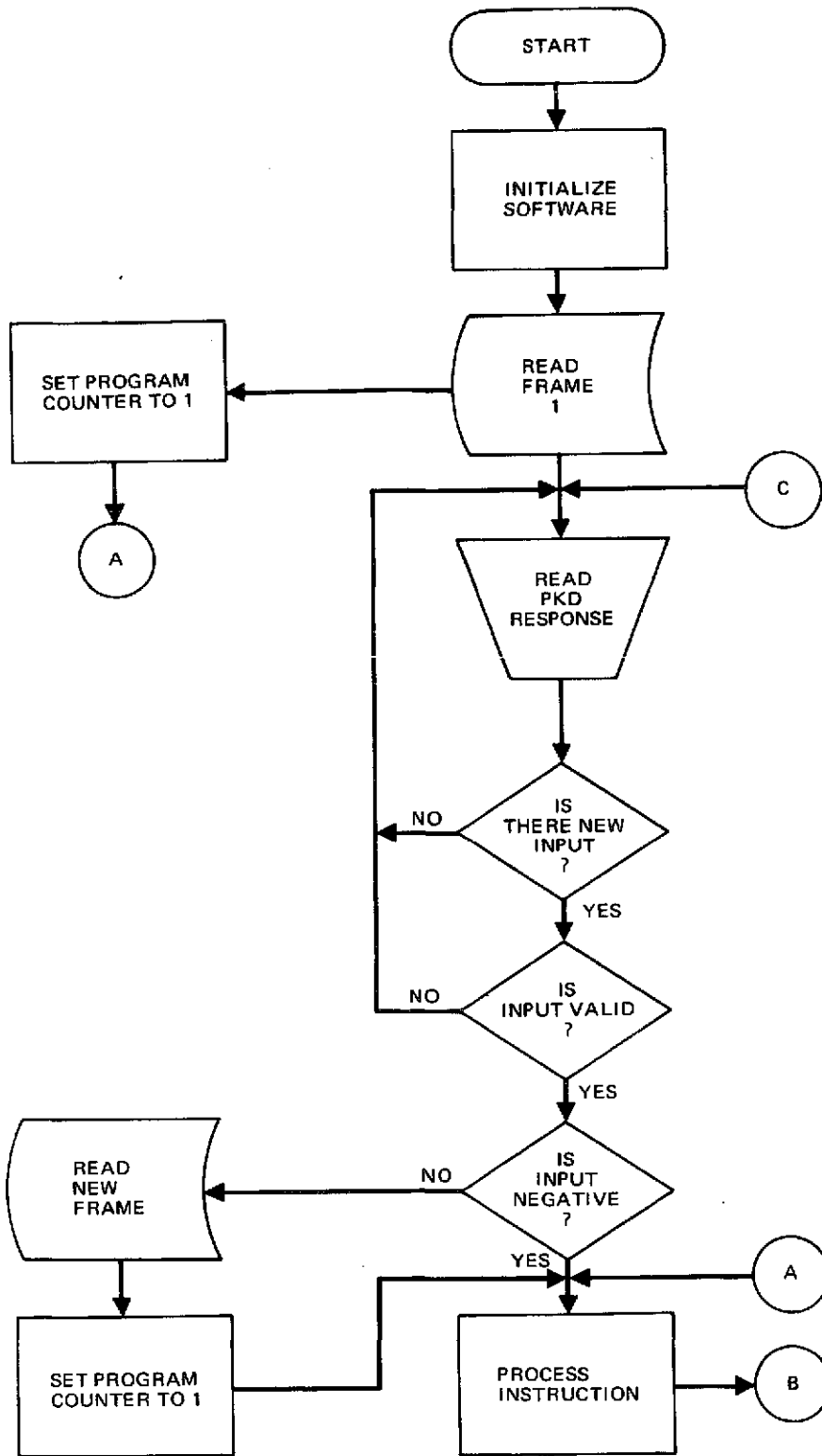
```
                        ┌──────────────┐
                        │    START     │
                        └──────────────┘
                               │
                               ▼
                        ┌──────────────┐
                        │  INITIALIZE  │
                        │   SOFTWARE   │
                        └──────────────┘
                               │
                               ▼
  ┌──────────────┐      ┌──────────────┐
  │ SET PROGRAM  │◄─────│    READ      │
  │ COUNTER TO 1 │      │   FRAME      │
  └──────────────┘      │     1        │
        │               └──────────────┘
        ▼                      │              ( C )
      ( A )                    ▼
                        ┌──────────────┐
                        │    READ      │
                        │    PKD       │
                        │  RESPONSE    │
                        └──────────────┘
                               │
                               ▼
                          ╱IS  ╲
                   NO   ╱THERE NEW╲
                  ◄────╱  INPUT    ╲
                       ╲    ?     ╱
                        ╲        ╱
                            │ YES
                            ▼
                          ╱  IS  ╲
                   NO   ╱ INPUT   ╲
                  ◄────╱  VALID    ╲
                       ╲    ?     ╱
                        ╲        ╱
                            │ YES
                            ▼
  ┌──────────────┐      ╱   IS   ╲
  │    READ      │ NO  ╱  INPUT   ╲
  │    NEW       │◄───╱ NEGATIVE   ╲
  │   FRAME      │    ╲     ?      ╱
  └──────────────┘     ╲         ╱      ( A )
        │                 │ YES
        ▼                  ▼
  ┌──────────────┐   ┌──────────────┐
  │ SET PROGRAM  │   │  PROCESS     │──► ( B )
  │ COUNTER TO 1 │   │ INSTRUCTION  │
  └──────────────┘   └──────────────┘
```
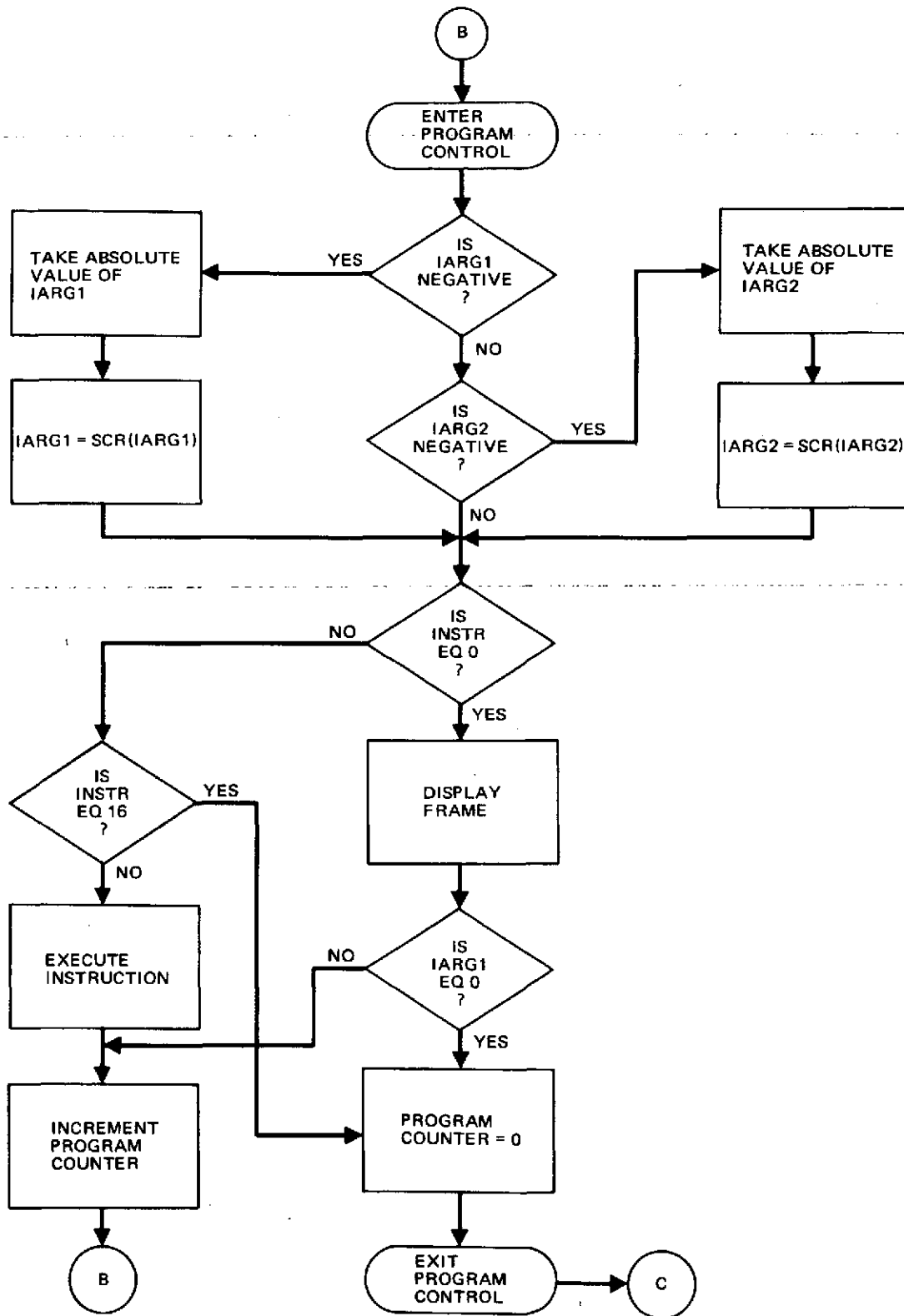
Figure C-3 (Page 1 of 2). PKD Executive Flow Chart

```
                              ( B )
                               │
                               ▼
                        ┌──────────────┐
                        │    ENTER     │
                        │   PROGRAM    │
                        │   CONTROL    │
                        └──────────────┘
                               │
                               ▼
┌──────────────┐          ◇ IS ◇          ┌──────────────┐
│ TAKE ABSOLUTE│◄── YES ──│ IARG1 │──────►│ TAKE ABSOLUTE│
│ VALUE OF     │          │NEGATIVE│       │ VALUE OF     │
│ IARG1        │          │   ?   │        │ IARG2        │
└──────────────┘          └───────┘        └──────────────┘
       │                     │ NO                 │
       ▼                     ▼                     ▼
┌──────────────┐          ◇ IS ◇          ┌──────────────┐
│IARG1 = SCR   │          │ IARG2 │── YES►│IARG2 = SCR   │
│   (IARG1)    │          │NEGATIVE│       │   (IARG2)    │
└──────────────┘          │   ?   │        └──────────────┘
       │                     │ NO                 │
       └─────────────────────┼────────────────────┘
                             ▼
                         ◇ IS ◇
              NO ────────│ INSTR │
              │          │ EQ 0  │
              │          │   ?   │
              ▼             │ YES
          ◇ IS ◇           ▼
          │ INSTR │── YES  ┌──────────┐
          │ EQ 16 │───┐    │ DISPLAY  │
          │   ?   │   │    │  FRAME   │
          └───────┘   │    └──────────┘
             │ NO     │         │
             ▼        │         ▼
      ┌──────────┐    │     ◇ IS ◇
      │ EXECUTE  │    │ NO──│ IARG1 │
      │INSTRUCTION│   │     │ EQ 0  │
      └──────────┘    │     │   ?   │
             │        │        │ YES
             ▼        │        ▼
      ┌──────────┐    │   ┌──────────┐
      │INCREMENT │    └──►│ PROGRAM  │
      │ PROGRAM  │        │COUNTER=0 │
      │ COUNTER  │        └──────────┘
      └──────────┘            │
             │                ▼
           ( B )          ┌──────────┐
                          │  EXIT    │──►( C )
                          │ PROGRAM  │
                          │ CONTROL  │
                          └──────────┘
```

Figure C-3 (Page 2 of 2). PKD Executive Flow Chart

Table C-1

EXECUTION CONTROL PROGRAM OPERATIONS

Operator	Program Array (314) P(N, 1)=CODE	P(N, 2)=ARG1	P(N, 3)=ARG2	Description
DISPLAY	0	N	—	Output display, wait for PKD input if N=0; else, continue to next program element
REPLACE LINE	1	N1	N2	Line(N2)→Line(N1)
REPLACE SECTOR	2	N1	N2	Sector(N2)→Sector(N1) Command(N2)→Command(N1)
REPLACE COMMAND	3	N1	I	I→Command(N1)
REPLACE FUNCTION	4	N1	I	I→Function(N1)
KEYBOARD INPUT	5	(Row)	(Column)	Cursor moved to indicated row and column. Number input→SCR(1)*
ADD	6	N1	N2	SCR(N2)+SCR(N1)→SCR(N2)
SUBTRACT	7	N1	N1	SCR(N2)-SCR(N1)→SCR(N2)
MULTIPLY	8	N1	N1	SCR(N2)*SCR(N1)→SCR(N2)
DIVIDE	9	N1	N1	SCR(N2)/SCR(N1)→SCR(N2)
MOVE	10	N1	N2	SCR(N2)→SCR(N1)
DISPLAY REAL	11	(Row)	(Column)	(SCR(1)) displayed at indicated row and column
FRAME	12	N1	N2	Get Display N1, if not in file skip instruction. If N2≠0, save current display number as N2
CODE	13	ARG1	ARG2	Link to application programs arguments as required
GO TO	14	N1	—	Execute instruction at Location N1
GENERATE REAL	15	(Row)	(Column)	(SCR(1)) moved to display at indicated row and column
END	16	—	—	

*SCR=Scratch (F10,3)

A negative IARG1 or IARG2 contained in a PKD instruction results in the argument being replaced by SCR(IARG1 or IARG2) at execution.

The first three locations of the scratch table are utilized by the executive. SCR(1) contains the floating point number from the last KYBIN call. SCR(2) contains the current PKD frame in core. SCR(3) contains the number of the last sector called.

Program table location 1 is reserved for the instruction executed after reading a new frame into the core.

APPENDIX D
EXPERIMENT SCENARIO

Appendix D

EXPERIMENT SCENARIO

The following section provides a capsule explanation of the relationship
between the frames when they are grouped into activity phases. Section D. 2
contains all of the PKD display frames defined during the analysis of the
control functions required by the Crop Resource Experiment, S192.

D. 1 EXPLANATORY NOTES

Starting with the top-level crop resources experiment control frame (104),
block diagrams illustrating the sequence of displays for each experiment
activity are presented in Figures D-1 through D-8. The activities correspond
to the second-level block titles of Figure D-1. Whenever an activity is com-
plated or terminated, the program returns to Frame 104 to permit selection
of the next task. Frames in Section D. 2 prior to 104 are included as typical
displays required to select Experiment S192.

The following sections discuss the experimenter's role in the performance
of each activity outlined in Figures D-2 through D-8.

D. 1. 1 Experiment Planning (Figure D-2)

    105 - Planning

            Select planning interest

    106 - Status

            For a given experiment number, a CRT display of overall
            status is called up for review or analysis

    108 - Schedule

            For a given experiment number, a CRT display of the current
            schedule is called up for review or analysis

```
                                         104 ┌──────────────┐
                                             │ SELECT       │
                                             │ EXPERIMENT   │
                                             │ PHASE        │
                                             └──────────────┘
```

| 105 EXPERIMENT PLANNING | 121 POWER UP | 125 BEGIN OPERATIONS (STARTUP) | 141 RUN EXPERIMENT | 146 TERMINATE | 151 DATA ANALYSIS | 180 CHECKOUT |
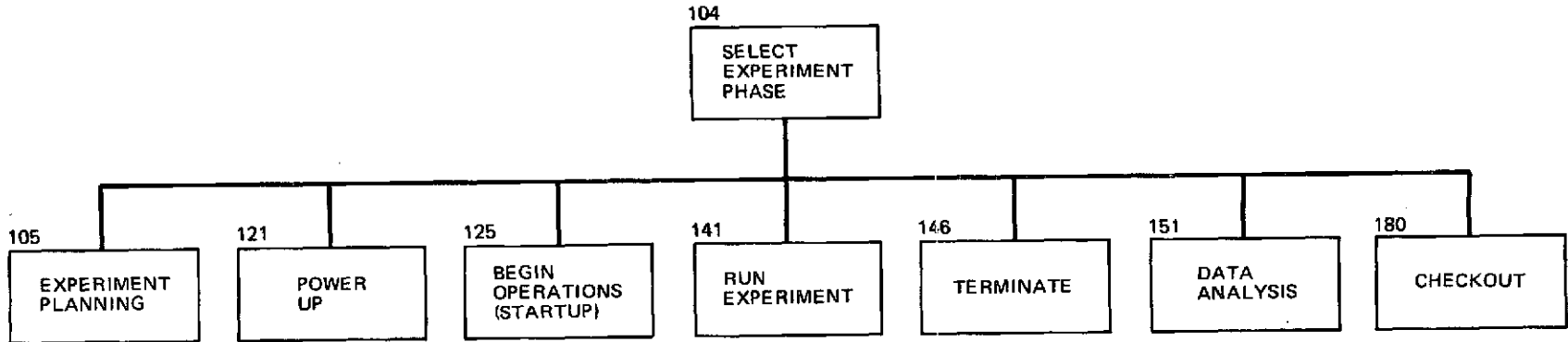
Figure D-1. Sequence of Displays

```
                        106
                    ┌──────────────────┐
                    │  PLANNING        │
                    │  SELECT PLANNING │
                    │  INTEREST        │
                    └──────────────────┘
```

106
┌──────────────────┐
│ STATUS SELECT   │
│ EXPERIMENT; CRT │
│ DISPLAY OF STATUS│
└──────────────────┘

108
┌──────────────────┐
│ SCHEDULE   │
│ SELECT EXPERIMENT -│
│ CRT DISPLAY OF   │
│ SCHEDULE   │
└──────────────────┘

111
┌──────────────────┐
│ UPDATE   │
│ SELECT EXPERIMENT│
│ AND PROCEED   │
└──────────────────┘

112
┌────────────────────────┐
│ CURRENT PLANS   │
│ COMPLETE EXPERIMENT;   │
│ EXAMPLE - ABORT, CON- │
│ TINGENCY, OR RETURN   │
└────────────────────────┘

113
┌────────────────────────┐
│ CURRENT PLAN   │
│ INCOMPLETE EXPERI-   │
│ MENT. EXAMPLE -   │
│ ABORT, CONTINGENCY,   │
│ PLAN, OR RETURN   │
└────────────────────────┘

181
┌──────────────────┐
│   CHECKOUT   │
└──────────────────┘

114
┌────────────────────────┐
│ CONTINGENCY   │
│ TYPICAL PLANNED   │
│ CONTINGENCY -   │
│ OPTIONAL EXIT TO   │
│ CHECKOUT   │
└────────────────────────┘

Figure D-2. Experiment Planning

121
┌──────────────────┐
│   POWER UP   │
└──────────────────┘

122
┌──────────────────┐
│   CHECKLIST   │
└──────────────────┘

123
┌──────────────────┐
│   SETUP   │
└──────────────────┘

124
┌──────────────────┐
│   SUMMARY   │
└──────────────────┘

(104)
┌──────────────────┐
│ EXPERIMENT   │
│ CONTROL   │
└──────────────────┘

Figure D-3. Power Up

```
  125                      126                     127                      128
┌──────────────────┐   ┌──────────────┐   ┌──────────────────┐   ┌──────────────┐
│ BEGIN OPERATIONS │───│   CHECKLIST  │───│      SETUP       │   │   CH SELECT  │
└──────────────────┘   └──────────────┘   └──────────────────┘   └──────────────┘
```
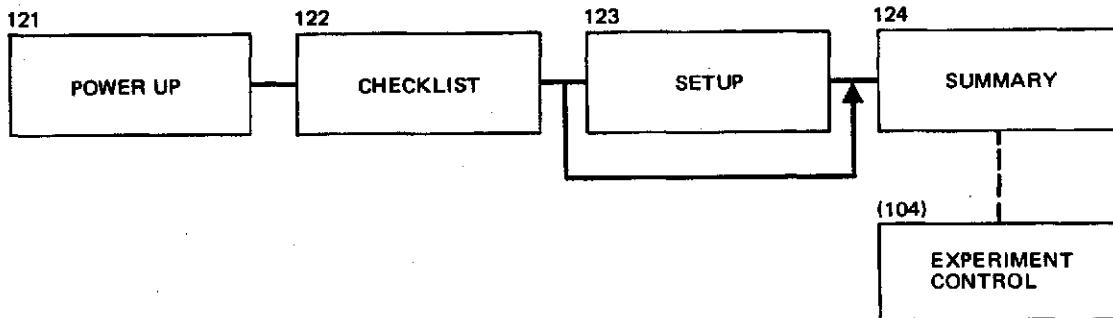
Figure D-4. Begin Operations

```
                    141
           ┌──────────────────┐
           │  RUN EXPERIMENT  │
           └──────────────────┘

  142                          143
┌──────────────┐          ┌──────────────────┐
│  AUTOMATIC   │─────────▶│  MANUAL SEQUENCE │
│  SEQUENCE    │          │                  │
└──────────────┘          └──────────────────┘

           (146)
           ┌──────────────────┐
           │    TERMINATE     │
           └──────────────────┘
```
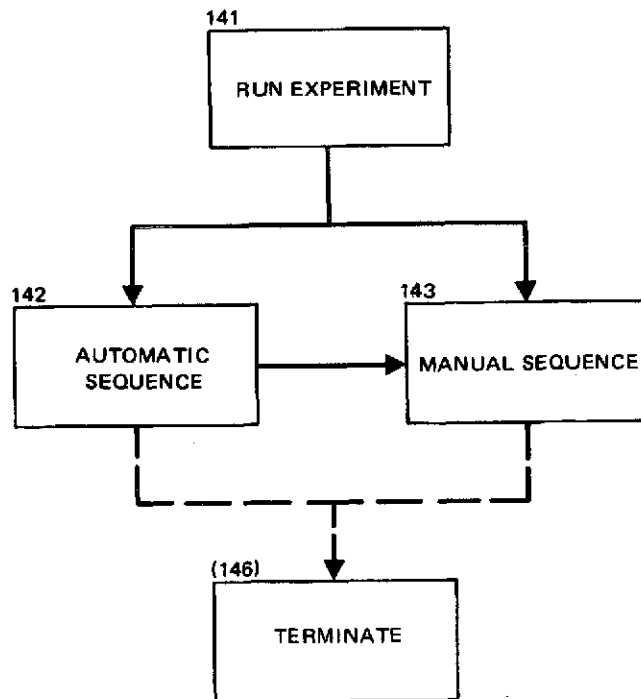
Figure D-5. Run Experiment
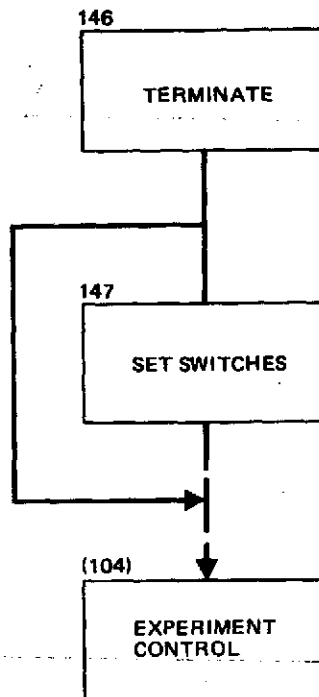
Figure D-6. Terminate

111 - Update

Indicate the experiment which is to be updated

112 - Updated Current Plan

Typical presentation for completely planned experiment.
Contingency planning or aborting the experiment can occur
at this point.

113 - Current Plan

Example of incomplete plans. Operator completes planning
by selecting channels or invokes planned contingencies or
aborts.

114 - Contingency

Those contingencies which can be anticipated and circumvented
by preplanned solutions are presented for selection. If backup
equipment is to be brought on-line, the checkout phase can be
entered from this level.

151
DATA ANALYSIS
INPUT GROUND TRUTH
TRAIN CLASSIFIER,
OPTIMIZE MSS
CHANNEL USAGE

152
TRUTH SITE
INPUT EXISTING
GROUND TRUTH DATA

161
TRAINING
COMMIT TO COMPUTER
RUN-ANALYZE
OBSERVED DATA

171
OPTIMIZE
CLASSIFIER RESULTS
ACCEPT OR
RE-OPTIMIZE

153
FINE ADJUST
RECTIFY OBSERVED DATA
TO TRUTH SITE VIEW
LANDMARKS INPUT

162
RUN MODE
INITIAL RUN,
RE-OPTIMIZE OR
CONCENTRATE
OPTIMIZATION

154
CORRELATION
START COMPUTER RUN
(COMMIT RESOURCES)

163
CROP ANALYSIS
INITIAL OR RE-OPTIMIZE
CROP ID RUN — ENTER
NO. OF CHANNELS TO
TRAIN DISCRIMINATOR

263
CROP ANALYSIS
(WHEAT)
REF. 163

155
RESULTS
RUN COMPLETED
RETURN TO 151 FOR
NEXT ACTION —
TRAINING

165
INITIALIZE
DISCRIMINANT TRAINER
FOR SELECTED
CHANNELS

272
TRAINING
RESULTS
(WHEAT)
REF. 170

170
TRAINING RESULTS
REJECT RESULT AND
RETRAIN OR
CONTINUE WITH
NEXT CROP

86

Figure D-7. Data Analysis

Figure D-8. Checkout

D. 1. 2   Power Up (Figure D-3)

121 - Power Up

Lists equipment used in the experiment and indicates overall status. The experimenter proceeds to make a checklist of the operations.

122 - Checklist

Experiment-related switch settings are presented for review. If checklist is acceptable, next display is 124.   If switch settings are to be changed, Frame 123 is called.

123 - Setup

Operator sends commands or indicates manual switch settings by pressing the appropriate display switch.

124 - Summary

Display of final instructions to the operator. Current timeline data is included to guide the operator in selecting a subsequent activity. Returns to Frame 104 in normal sequence.

D. 1. 3 Begin Operations (Figure D-4)

125 - Begin Operation

Equipment list and timeline data are displayed along with pre-operation preparation status.

126 - Checklist

Similar to Frame 122, except timeline data are included.

127 - Setup

Same as 123.

128 - Channel Selection

Operator performs final equipment setup prior to running experiment.

D. 1. 4 Run Experiment (Figure D-5)

141 - Run Control

Select between automatic and manual sequence at operator's discretion. Starting and current times are displayed. Operator may override an automatic inhibit condition when present.

142 - Automatic Sequence

Timeline data and manual switchover provided. Quick-look data can be requested for display.

143 - Manual Sequence

Similar to 142 except switchover from manual to automatic is not allowed.

D. 1. 5 Terminate (Figure D-6)

146 - Terminate

This display is automatically called at the end of the automatic or manual sequence. The option is to accept the current switch settings, in which case return is to Frame 104 or to set switches in Frame 147 before return.

D. 1. 6 Data Analysis (Figure D-7)

151 - Data Analysis

Select from Input Ground Truth, Train, or Optimize. (Note: Train will not be accepted until Ground Truth Site data have been input - Frame 162 is displayed. Optimize will not be accepted until the classifier has been trained - Frame 172 is displayed.)

152 - Truth Site

An accurate description of the ground truth site is used to train the classifier. These data are input from tape at this time to be compared to the multispectral scanner data acquired during the mission. An initial evaluation of the scanner channels is performed by comparing the known truth site crops and their location to the acquired data by pattern recognition techniques. The result is initial crop identification by multispectral scanner channel output. The comparative channel data are used in the training phase to select the best channels for crop identification (see Frame 163).

The commands for positioning the tape are displayed in Frame 152. Visual observation on a CRT would accompany the PKD display.

153 - Fine Adjust

A photograph of the truth site is used to locate the position of landmarks. The CRT light pen is then used to input the coordinates of each landmark.

154 - Correlation

Review time and computer resources required to run correlation program before starting computer run.

155 - Results

Tutorial frame - no action required.

161 - Training

Commit to computer run which will train classifier by analyzing observed data versus ground truth data.

162 - Run Computer

During training, initial run must be selected. Re-optimize is the proper command when in optimize mode or for retraining. For this demonstration, the separability level is fixed.

163 - Result (Most Identifiable Crop - Soil)

Results of the computer run are presented. The scanner channels are ranked according to their accuracy in identifying the crop "soil" using the pattern recognition technique of comparing channel output during truth site flyover to the known layout of the truth site, as defined by the taped data input in Frame 152. The operator inputs the lowest number of channels which can identify the crop within the accuracy required by the experiment.

165 - Initialize (2 channels - Soil)

Initialization of discriminator when two channels are used. Accompanies CRT display of data points between the two channels. The straight line defined by the two points input is the starting point for the discriminator. By allowing the

experimenter to define an initial line of separation between
soil data and other crops, the computer run time is reduced
to an acceptable amount. Without the experimenter's input,
the discriminator could try an enormous number of random
lines before converging on the optimum line.

170 - Training Results (1 channel - Soil)
Experimenter accepts or rejects percent of false classification
from results of training the classifier discriminant for the
selected number of channels.

270 - Training Results (2 channels - Soil)
Typical results - similar to Frame 170, except result is for
two channels.

263 - Results (Second Most Identifiable Crop - Wheat)
Same as 173 except crop is wheat.

272 Training Results (1 channel - Wheat)
Same as 170 except crop is wheat.

171 - Optimize
After training, experimenter can reoptimize by reentry to the
training procedure via this frame. In this mode, the reoptimize
option of Frame 162 is enabled.

D.1.7 Checkout (Figure D-8)

180 - Checkout
Select instrument to be checked from list of experiment-
related equipment.

181 - Multispectral Scanner
Computer playback of input for operator's review. Entry is
from either 180 or from Contingency Planning (114).

182 - Check

Operator initiates check mode if timeline indicates sufficient time is available.

183 through 186 - Verify Results

Operator verifies results of checkout procedure data.

187 - Operation Complete

Typical successful checkout. Timeline is repeated to allow operator to select next activity. PROCEED will return to Frame 104.

188 - Fault Detected

Typical failure display. Checkout can be continued on another instrument (do next checkout), replanning can be entered (replan experiments), or a replacement part can be installed.

189 - Replace

Typical checklist of a part replacement procedure. Operator indicates performance of each step to computer. PROCEED completes checkout.

D. 2 SUMMARY OF THE DISPLAYS FOR THE EXPERIMENT CONTROL SCENARIO

D. 2. 1 Introductory Frames - 2 through 104

PKD APPLICATION DEFINITION

Frame number (I5) ___6___

```
 1 |   |   |   |   | E | X | P | E | R | I | M | E | N | T |   | C | O | N | T | R | O | L |   |   |   |   |   |   |
 2 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
 3 | G | E | N | E | R | A | L |   | D | E | M | O | N | S | T | R | A | T | I | O | N |   | R | U | N |   |   |
 4 |   |   |   |   |   |   |   | I | N | T | R | O | D | U | C | T | I | O | N |   |   |   |   |   |   |   |   |
 5 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
 6 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
 7 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
 8 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
 9 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
10 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
```

Frame number (I5) _____2_____

1	M I S S I O N	F U N C T .	I N T E R A C .
2	P H A S E	C A T	T - L
3			
4			
5			
6			
7	A C T I O N :	S E L E C T	M O D E
8			
9			
10			

Frame number (I5) _____3_____

1	M I S S I O N	D A T A	C O M M	C H E C K
2	C O N T R O L	M G M T		O U T
3	M N V R .		F L I G H T	
4	M G M T	G & N	C O N T R O L	E X P
5	C R E W			
6	F U N C T .			
7	A C T I O N :	S E L E C T	C A T E G O R Y	
8				
9				
10				

94

Frame number (I5) __101__

1. ACTION: SELECT FUNCTIONAL
2. PROGRAM ELEMENT (FPE)
3.
4.
5. ASTRON PHYSICS   EARTE   COM/
6.   SURVEY   NAV
7. MATRL / TECHNOL   LIFE
8. MANUFCT   SCIENCE
9.
10.

Frame number (I5) __102__

1. ACTION: SELECT SUBPROGRAM ELEMN
2.
3. WEATH +   LAND   AIR+H2O   RESOURCE
4. ATMOS.   USE   POLLUT.   RECOGNI
5.
6.
7. NATURL   OCEAN   MINIMUM
8. DISASTR   RESOURCE   PAYLOAD
9.
10.

Frame number (I5) ___103___

| 1 | A | C | T | I | O | N | : | | S | E | L | E | C | T | | R | E | S | O | U | R | C | E | | S | U | R | V | E | Y | |

| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 3 | | C | R | O | P | S | | | | F | O | R | E | S | T | | | W | A | T | E | R | | | M | I | N | E | R | A | L | S |

| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Frame number (I5) ___104___

| 1 | A | C | T | I | O | N | : | | S | E | L | E | C | T | | E | X | P | E | R | I | M | E | N | T | | P | H | A | S | E |

| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 3 | P | L | A | N | S | | + | | | P | O | W | E | R | | | C | H | E | C | K | | | B | E | G | I | N | |

| 4 | | U | P | D | A | T | E | | | | U | P | | | | | O | U | T | | | | | O | P | S | . | |

| 5 | | | R | U | N | | | T | E | R | M | N | A | T | E | | D | A | T | A | | | D | A | T | A | |

| 6 | | E | X | P | E | R | . | | | O | P | S | . | | | A | N | A | L | Y | S | I | S | | T | R | A | N | S | . |

| 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 9 | K | E | Y | | I | N | | E | X | P | I | M | E | N | T | | # | | | | | . | | | | | | | |

| 10 | C | O | N | T | R | O | L | | O | F | | E | X | P | . | | # | | | | | . | | | | | | | |

D.2.2 Plans and Update Frames - 105 through 114

PKD APPLICATION DEFINITION

Frame number (I5) ___105___

```
1  | A C T I O N :   S E L E C T   P L A N N I N G   I N T E R E S
2  |
3  | C U R R E N T     C U R R E N T     P L A N     U P D A T E
4  |   S T A T U S   S C H E D U L E   C R E A T E     /   A B O R
5  |
6  |
7  |
8  |
9  |
10 |
```

PKD APPLICATION DEFINITION

Frame number (I5) ___106___

```
1  | A C T I O N :   F O R   D E T A I L E D   S C H E D U L E
2  | A N D   S T A T U S   O N   C R T   K E Y   I N   T H E
3  | E X P E R I M E N T   #
4  |
5  | P R E S S   T O   R E C A L L   O V E R A L L :   C U R R E N T
6  |                                                     S T A T U S
7  |
8  | T O   C O N T I N U E   P L A N N I N G   P R E S S   P R O .
9  |                                                     N O T
10 |                                                 A V A I L A
```

97

PKD APPLICATION DEFINITION

Frame number (I5) ___108___

1. | A | C | T | I | O | N | : | | F | O | R | | D | E | T | A | I | L | E | D | | S | C | H | E | D | U | L | E | | |
2. | A | N | D | | S | T | A | T | U | S | | O | N | | C | R | T | | K | E | Y | | I | N | | T | H | E | | |
3. | E | X | P | E | R | I | M | E | N | T | | # | | | | | | | | | | | | | | | | | | |
4. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
5. | P | R | E | S | S | | T | O | | R | E | C | A | L | L | | O | V | E | R | A | L | L | | | C | U | R | R | E | N | T |
6. | | | | | | | | | | | | | | | | | | | | | | | | | | S | C | H | E | D | U | L | E |
7. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
8. | T | O | | C | O | N | T | I | N | U | E | | P | L | A | N | N | I | N | G | | P | R | E | S | S | | P | R | O | . |
9. | | | | | | | | | | | | | | | | | | | | | | | | | | N | O | T | | |
10. | | | | | | | | | | | | | | | | | | | | | | | | A | V | A | I | L | . | |

PKD APPLICATION DEFINITION

Frame number (I5) ___111___

1. | A | C | T | I | O | N | : | | K | E | Y | | I | N | | E | X | P | . | | # | | | | | | | | | |
2. | P | R | E | S | S | | P | R | O | . | | T | O | | S | T | A | R | T | | U | P | D | A | T | E | | | | |
3. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
4. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
5. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
6. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
7. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
8. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
9. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
10. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

PKD APPLICATION DEFINITION

Frame number (I5) ___112___

1	EXP #	UPDATE	SELECT
2	CURRENT PLANS	COMPLETE	OPTION
3		PRESS	INTERACT
4		TO	TIMELINE
5		TAKE NO	ALLOWED
6		ACTION	CONTIN.
7			EXP
8			ABORT
9			
10			

PKD APPLICATION DEFINITION

Frame number (I5) ___113___

1	EXP #	UPDATE	SELECT
2	CURRENT PLANS	REQUIRE:	OPTION
3	(1) KEY IN FOR MSS		INTERACT
4	REQ CHANNELS #	, # ,	TIMELINE
5	# ,	# , # ,	ALLOWED
6			CONTIN.
7	(2) PRESS	PLAN 99%	EXP
8		COMPLETE COMPLT	ABORT
9			
10			

Frame number (I5) ___114___

```
1   | A C T I O N :   I N D I C A T E   C O N T I N G E N C Y |
2   |                                                         |
3   | D E L A Y   B Y                                         |
4   |   C L O U D S                                           |
5   |                                                         |
6   |                                                         |
7   | C A L   S O U R C E                                     |
8   | F A I L U R E                                           |
9   | R E S C H E D .       T O       R E M E D Y   R E M E D Y |
10  | #           #               N O W         L A T E R     |
```

D. 2. 3 Power-Up Frames - 121 through 124

Frame number (I5) ___121___

1	P	O	W	E	R	-	U	P		P	R	O	C	E	D	U	R	E		F	O	R		E	X	P						
2	I	N	S	T	R	U	M	E	N	T	S		T	O		B	E		A	C	T	I	V	A	T	E	D		A	R	E	
3																															
4					M	U	L	T	I	S	P	E	C	T	R	A	L		S	C	A	N	N	E	R							
5																															
6	A	L	L		P	O	S	T		L	A	U	N	C	H		P	R	E	P	S	.		C	O	M	P	L	E	T	E	
7	A	C	T	I	O	N	:		P	R	E	S	S		P	R	O	C	E	E	D		F	O	R								
8			-								C	H	E	C	K		L	I	S	T												
9																															
10																															

Frame number (I5) ___122___

1	A	C	T	I	O	N	:		C	O	N	F	I	R	M					O	R				S	E	Q	U	E	N	C	E
2								C	H	E	C	K		L	.								S	W	I	T	C	H	E	S		
3	C	A	L		S	O	U	R	C	E		S	1				S	2					S	3								
4	I	N	T	E	N	S	I	T	Y		H	I	G	H				H	I	G	H					L	O	W				
5		M	O	D	E			S	T	A	N	D	B	Y																		
6							M	O	D	E																					
7	P	O	W	E	R			P	O	W	E	R																				
8									O	N																					
9																															
10																															

Frame number (I5) 123

1	A C T I O N :	S E T	S W I T C H E S	T H E N		
2		P R E S S	P R O C E E D			
3	S 1	S 1		S 2	S 2	
4	H I G H	L O W		H I G H	L O W	
5	S 3	S 3		R E A D Y	S T A N D B Y	
6	H I G H	L O W		M O D E	M O D E	
7		P O W E R		P O W E R		
8		O N		O F F		
9						
10						

Frame number (I5) 124

1	P O W E R O N	M U L T I S P E C T R A L	S C A N N E R
2	T I M E I S 7 2 . 1 2	3 0 M I N .	W A R M - U P
3			
4			
5	N E X T R E Q U I R E D	A C T I O N I S	
6	C H E C K - O U T (C A L) A T 7 2 . 4 5		
7			
8	P R E S S P R O C E E D		
9	M U L T I S P	M U L T I S P	M U L T I S P
10	7 2 . 7 0	7 3 . 6 5	8 2 . 7 0

D.2.4 Begin Operations Frames - 125 through 128

PKD APPLICATION DEFINITION

Frame number (I5) ___125___

```
1  | B E G I N   O P E R A T I O N S   F O R   E X P
2  | I N S T R U M E N T S   T O   B E   O P E R A T E D   A R E :
3  |         M U L T I S P E C T R A L   S C A N N E R
4  |
5  | T I M E   I S   T I M E   R E Q               E X P   R U N
6  |   7 3 . 0 5 0   I S   0 . 0 6 0               A T   7 3 . 1 1
7  | A L L   P R E O P   P R E P A R A T I O N S   C O M P L E T E
8  | A C T I O N :   P R E S S   P R O C E E D   T O   B E G I N
9  |
10 |
```

PKD APPLICATION DEFINITION

Frame number (I5) ___126___

```
1  | A C T I O N :   C O N F I R M       O R     S E Q U E N C E
2  |             C H E C K   L .             S W I T C H E S
3  |   C H E C K       D O O R       T A P E S   S P E E D
4  |     M O D E       O P E N       1 , 2 , 3   6 0   I P S
5  | R E C O R D I N G   A L L
6  | C H A N N E L S
7  | 6   M I N ( 0 . 0 6 ) F O R   E L E C T .   S T A B I L I Z E
8  | N E X T   R E Q U I R E D   A C T .   I S   R U N   E X P E R .
9  |                             T A P E S   S P E E D
10 |                       /         1 5   I P S
```

103

Frame number (I5) __127__

1	ACTION: SET SWITCHES
2	PRESS PROCEED WHEN DONE
3	CHECK DOOR TAPE SPEED
4	MODE OPEN ~MO. 1 7.5 IPS
5	STANDBY DOOR TAPE
6	MODE CLOSED NO. 2 15 IPS
7	READY TAPE
8	MODE NO. 3 60 IPS
9	NO. 1 NO. 2 NO. 3
10	ASSIGNED ASSIGNED ASSIGNED

Frame number (I5) __128__

1	ACTION: REVISE CHANNELS OR/THEN
2	PRESS PROCEED
3	RECORDING ALL
4	CHANNELS
5	DELETE # , , , DELETE
6	, , , , , ; ALL
7	ADD # , , , , , , ADD
8	, , , , , : ALL
9	
10	

D.2.5 Run Experiment Frames - 141 through 143

Frame number (I5) __141__

1. RUN CONTROL FOR EXPERIMENT .
2. ACTION: SELECT OPERATING MODE
3.
4.
5. AUTO   MANUAL
6. SEQUENCE   SEQUENCE
7. TIME IS EXP RUN
8. 73.101 AT 73.11
9. RUN CONTROL INHIBITED   OVERRIDE
10. STABILIZATION REQ+.05   INHIBIT

PKD APPLICATION DEFINITION

Frame number (I5) __142__

1. AUTOMATIC SEQUENCE IN CONTROL
2. ACTION: MONITOR EXP OR RETURN
3. TIME IS   OR   MANUAL
4. 73.102   SEQUENCE
5. START AT   DISPLAY VIDEO ON
6. 73.110   CHANNEL #
7. STOP AT
8. 73.115
9.
10.

Frame number (I5) ___143___

1	M	A	N	U	A	L		E	X	P		S	E	Q	U	E	N	C	E		I	N		C	O	N	T	R	O	L		
2	A	C	T	I	O	N	:		C	O	N	T	R	O	L		E	X	P	E	R	I	M	E	N	T					
3	T	I	M	E		I	S																									
4		7	3	.	1	0	3																									
5	S	T	A	R	T		A	T		P	R	E	S	S																	
6		7	3	.	1	1	5			S	T	A	R	T																	
7	S	T	O	P		A	T											D	I	S	P	L	A	Y		V	I	D	E	O		O	N
8		7	3	.	1	1	5										C	H	A	N	N	E	L		#						
9		T	I	M	E		I	S									P	R	E	S	S				T	I	M	E		I	S
10		7	3	.	0	0	9										S	T	O	P				7	3	.	1	1	6	

D. 2. 6 Terminate Operations Frames - 146 and 147

PKD APPLICATION DEFINITION

Frame number (I5) ___146___

```
1  | T E R M I N A T I O N   O F   O P S   F O R   E X P
2  | A C T I O N :   C O N F I R M       O R
3  |               C H E C K                       S E T
4  |               L I S T                 S W I T C H E S
5  | S T A N D B Y
6  |     M O D E
7  |     D O O R       P O W E R
8  |   C L O S E D       O F F
9  |
10 |
```

PKD APPLICATION DEFINITION

Frame number (I5) ___147___

```
1  | A C T I O N :   S E T   S W I T C H E S   T H E N   P R E S S
2  |               P R O C E E D
3  | S T A N D B Y     C H E C K     R E A D Y
4  |     M O D E       M O D E       M O D E
5  |     D O O R       D O O R
6  |   C L O S E D     O P E N
7  |     P O W E R     P O W E R
8  |       O F F         O N
9  |
10 |
```

D.2.7 Data Analysis Frames - 151 through 272

PKD APPLICATION DEFINITION

Frame number (I5) __151__

```
1  | D A T A   A N A L Y S I S   E X P E R .   #
2  | A C T I O N :   S E L E C T   T Y P E   O F   A N A Y S I S
3  |
4  |
5  | E X A M I N E   V E R I F Y   E D I T   I N P U T
6  |   D A T A   Q U A L   / F I L E   G N D   T R U E
7  |   T R A I N   O P T I M I Z E   O N B O A R D
8  |
9  |
10 |
```

PKD APPLICATION DEFINITION

Frame number (I5) __152__

```
1  | A C T I O N :   O B T A I N   T R U T H   S I T E   P H O T O
2  | O V E R L A Y   F R O M   F I L E   F 1 2 3 4 . 5   T H E N
3  | C Y C L E   D A T A   T A P E S   O N   C R T   T O   M A T C H
4  | K E Y   I N   C H A N N E L   # =   F O R   D I S P L A Y
5  | F O R W A R D   S T O P   R E V E R S E
6  |
7  |   H I G H   L O W   M A T C H
8  |   S P E E D   S P E E D   F A I R
9  |
10 |
```

108

Frame number (I5) 153

1	ACTION: FINE ADJUST TRUTH PHOTO
2	OVERLAY TO RECTIFIED OBS DATA
3	THEN USE LIGHT PEN TO KEYIN LAND
4	MARKS INDICATED ON PHOTO.
5	LIGHT PEN PRESS PROCEED
6	IS ON TO CONTINUE
7	LAND LAND LAND
8	MARK 1 MARK 2 MARK 3
9	X1=0287 X2=0431 X3=0668
10	Y1=0481 Y2=0259 Y3=0592

PKD APPLICATION DEFINITION

Frame number (I5) 154

1	COMPUTER CORRELATING TRUTH TO
2	OBS DATA COMP. TIME REQ = 5 MIN
3	CORE REQ = 57%
4	ACTION: SELECT OPTION
5	NEW EXP OTHER RUN DONT
6	PHASE DATA AN COMPUT RUN
7	
8	
9	
10	

Frame number (I5) ___155___

D	A	T	A		C	O	R	R	E	L	A	T	I	O	N		P	R	O	G	.	C	O	M	P	L	E	T	E

(rows 1–3, 5–10 blank)

Frame number (I5) ___161___

| C | O | M | P | U | T | E | R | | T | R | A | I | N | I | N | G | | O | F | | C | L | A | S | S | I | F | I | E | R |

| F | O | R | | S | E | L | E | C | T | . | | C | H | N | L | S | | O | N | | M | S | | S | C | A | N | N | E | R |

| R | U | N | | T | I | M | E | | 1 | 0 | | M | I | N | | | C | O | R | E | | ⇒ | | 3 | 5 | % |

(row 4 blank)

| A | C | T | I | O | N | : | | S | E | L | E | C | T | | O | P | T | I | O | N |

(row 6 blank)

| | | | | | | | | R | U | N | | | D | O | N | T | | R | U | N |

| | | | | | | | | C | O | M | P | U | T | E | R | | | C | O | M | P | . |

(rows 9–10 blank)

Frame number (I5) ___162___

| 1 | A | C | T | I | O | N | : | | S | E | T | | C | O | M | P | U | T | E | R | | R | U | N | | S | W | I | T | C | H | |

| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 3 | I | N | I | T | I | A | L | | | F | R | O | M | | | S | E | P | A | R | A | T | A | B | I | L | I | T | Y | |

| 4 | | R | U | N | | | | | | | | | | L | E | V | E | L | | # | | = | | 1 | | T | O | | 9 |

| 5 | | R | E | - | | | | | | | | | | | | | | | | | | | | | | | | | |

| 6 | O | P | T | I | M | I | Z | E | | | | | | | | | | | | | | | | | | | | | |

| 7 | C | O | N | C | A | T | . | | | F | R | O | M | | | S | E | P | A | R | A | T | A | B | I | L | I | T | Y | |

| 8 | O | P | T | I | M | I | Z | E | | | | | | L | E | V | E | L | | # | | = | | | | T | O | | 9 |

| 9 | | N | O | T | | | K | E | Y | | I | N | | | S | E | P | A | R | A | T | A | B | I | L | I | T | Y | |

| 10 | A | C | T | I | V | E | | | | | | | | L | E | V | E | L | | # | | = | | | | | | |

Frame number (I5) ___163___

| 1 | N | O | 1 | | M | O | S | T | | I | D | E | N | T | I | F | I | A | B | L | E | | C | R | O | P | | I | S | : |

| 2 | | | | | | | S | O | I | L | | (| S |) | | | | | | | | | | | | | | | |

| 3 | B | E | S | T | | O | B | S | | C | H | A | N | N | E | L | S | | A | R | E | | : | | | | | |

| 4 | | 1 | > | 12 | > | 1 | 1 | > | 8 | > | 9 | > | 1 | 0 | > | 7 | > | 6 | > | 5 | > | 4 | > | 3 | > | 2 | > | 1 | 3 | |

| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 6 | A | C | T | I | O | N | : | | # | | O | F | | C | H | A | N | N | E | L | S | | T | O | | T | R | A | I | N |

| 7 | | | | | | | C | L | A | S | S | I | F | I | E | R | | = | | | | | | | | | |

| 8 | P | R | E | S | S | | P | R | O | C | E | E | D | | T | O | | C | O | N | T | I | N | U | E | | | |

| 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Frame number (I5) ___165___

1	A C T I O N : I N I T I A L I Z E D I S C R I M I N A N T
2	T R A I N E R O N C R T . M A R K T W O
3	P O I N T S O F S T . L I N E S E P A R A T I N G
4	S O I L (S) V R S R E M A I N I N G (R)
5	L I G H T P E N C H A N N E L S
6	I S O N 1 2 V R S 1
7	P R E S S T O P O I N T P O I N T C Y C L E
8	C O N T I N U E N O 1 N O 2
9	X 1 = 6 6 8 X 2 = 0
10	Y 1 = 7 6 0 Y 2 = 3 7 0 1 2 V R S 1 1

Frame number (I5) ___166___

1	A C T I O N : I N I T I A L I Z E D I S C R I M I N E N T
2	T R A I N E R O N C R T . M A R K T W O
3	P O I N T S O F S T . L I N E S E P A R A T I N G
4	W H E A T T Y P E 1 (W 1) V R S R E M A I N D E R (R)
5	L I G H T P E N C H A N N E L S
6	I S O N 1 1 V R S S
7	P R E S S T O P O I N T P O I N T C Y C L E
8	C O N T I N U E N O . 1 N O . 2 O F
9	X 1 = 2 8 7 X 2 = 6 4 1 C H A N N E L S
10	Y 1 = 4 8 1 Y 2 = 0 9 3 1 1 V R S 1 2

Frame number (I5) ___168___

1

2

3 R E M A I N I N G | C R O P S | N O T | P R O G R A M M E D

4 F O R | T H I S | D E M O N S T R A T I O N .

5 P R E S S | P R O C E E D | T O | C O N T I N U E | A S | I F

6 T H E | C L A S S I F I E R | H A D | B E E N | T R A I N E D

7 F O R | A L L | C R O P S

8

9

10

Frame number (I5) ___170___

1 O P T I M A L L Y | T R A I N E D | C L A S S I F I E R | F O R

2 S O I L | C H A N N E L S | 0 1

3 A C T I O N : | C H E C K | R E S U L T S | / | A P P R O V E

4

5 % | O B S | F A L S E L Y | C L A S S I F I E D | = | 1 0 %

6

7 A C C E P T | R E J E C T

8 % | F A L S E | % | F A L S E

9

10

113

Frame number (I5) __171__

```
1   OVERALL  CLASSIFIER  PERFORMANCE
2   NET  CHANNELS  USED
3   ACTUAL  CHANNELS  USED
4
5
6   ACTION:  SELECT  OPTION
7             REOPTIM.   ACCEPT
8             CHANNELS   CHNLS.
9
10
```

Frame number (I5) __172__

```
1   COMPUTER  TRAINING  DISABLED  UNTIL
2   CHANNELS  OPTIMIZED  FOR  EACH  CROP
3
4         PRESS  PROCEED  TO  RETURN
5
6
7
8
9
10
```

PKD APPLICATION DEFINITION

Frame number (I5) __173__

1. COMPUTER OPTIMIZATION DISABLED
2. UNTIL GROUND TRUTH IS INPUT.

4. PRESS PROCEED TO RETURN

PKD APPLICATION DEFINITION

Frame number (I5) __263__

1. NO 2 MOST IDENTIFIABLE CROP IS
2. WHEAT TYPE 1 (W1)
3. BEST OBS CHANNELS ARE:
4. 9 > 11 > 12 > 8 > 17 10 > 7 > 6 > 5 > 4 > 3 > 2 > 1 3
6. ACTION: # OF CHANNELS TO TRAIN
7. CLASSIFIER =

Frame number (I5) ___270___

```
1   O P T I M A L L Y   T R A I N E D   C L A S S I F I E R   F O
2   S O I L - .   C H A N N E L S   0 1   1 2
3   A C T I O N :   C H E C K   R E S U L T S   /   A P P R O V E
4
5   %   O B S   F A L S E L Y   C L A S S I F I E D   =   5 %
6
7                               A C C E P T       R E J E C T
8                           %   F A L S E   %   F A L S E
9
10
```

PKD APPLICATION DEFINITION

Frame number (I5) ___272___

```
1   O P T I M A L L Y   T R A I N E D   C L A S S I F I E R   F O R
2   W H E A T   ( W 1 )   C H A N N E L S     0 9
3   A C T I O N :   C H E C K   R E S U L T S   /   A P P R O V E
4
5   %   O B S   F A L S E L Y   C L A S S I F I E D   =   5 %
6
7                               A C C E P T       R E J E C
8                           %   F A L S E   %   F A L S E
9
10
```

116

D.2.8 Checkout Frames - 180 through 189

Frame number (I5) ___180___

1	ACTION: SELECT INSTRUMENT TO BE
2	CHECKED OUT
3	
4	
5	M-SPECT IR SPECT M-SPECT MICRO W
6	CAMERA METER SCANNER R-METER
7	MICRO W
8	ALTIMETER
9	
10	

Frame number (I5) ___181___

1	CHECK-OUT PROCEDURE EXPER
2	INSTRUMENTS TO BE CHECKED ARE:
3	
4	MULTISPECTRAL SCANNER
5	
6	
7	
8	ACTION: PRESS PROCEED TO CONTINU
9	
10	

Frame number (I5) ___182___

```
1  MULTISPECTRAL SCANNER CHECK-OUT
2  ACTION: CONFIRM
3              CHECK
4              MODE
5  TIME IS              TIME REG
6    72.45              IS 0.10
7       NEXT REQUIRED ACTION IS
8          BEGIN OPS AT 73.050
9  TIME IS
10   72.905
```

Frame number (I5) ___183___

```
1  VERIFY EACH CALIBRATION READING
2                        OK    NOT OK
3    DETECTOR ARRAY T=    %   (44-60%)
4     TH CAL SOURCE1 I=    %   (   -   %)
5     TE CAL SOURCE2 I=    %   (   -   %)
6     VIZ CAL SOURCE3 I=    %   (   -   %)
7     VIZ CAL SORCE5 I=    %   (   -   %)
8
9
10
```

Frame number (I5) ___184___

1	V	E	R	I	F	Y		E	A	C	H		C	A	L	I	B	R	A	T	I	O	N		R	E	A	D	I	N	G	
2																				O	K									O	K	
3			A	G	C	1		V	O	L	T	A	G	E			V	=				%		(	4	0	-	9	9	%	)	
4			A	G	C	2		V	O	L	T	A	G	E			V	=				%		(	4	0	-	9	9	%	)	
5			A	G	C	3		V	O	L	T	A	G	E			V	=				%		(	4	0	-	9	9	%	)	
6			A	G	C	4		V	O	L	T	A	G	E			V	=				%		(	4	0	-	9	9	%	)	
7			A	G	C	5		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
8			A	G	C	6		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
9																															
10																															

Frame number (I5) ___185___

1	V	E	R	I	F	Y		E	A	C	H		C	A	L	I	B	R	A	T	I	O	N		R	E	A	D	I	N	G	
2																				O	K				N	O	T		O	K		
3			A	G	C	7		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
4			A	G	E	8		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
5			A	G	E	9		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
6		A	G	C	1	0		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
7		A	G	C	1	1		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
8		A	G	C	1	2		V	O	L	T	A	G	E			V	=				%		(	3	6	-	9	9	%	)	
9																															
10																															

PKD APPLICATION DEFINITION

Frame number (I5) __186____

| 1 | V | E | R | I | F | Y | | E | A | C | H | | C | A | L | I | B | R | A | T | I | O | N | | R | E | A | D | I | N | G | | |

Row 1: VERIFY EACH CALIBRATION READING

Row 2: OK   NOT OK

Row 3: AGC 13-1 VOLTAGE = % (36 - 98 %)

Row 4: AGC 13-2 VOLTAGE = % (36 - 98 %)

Row 5: CAL SOURC1 TEMP = % (82 - 86 %)

Row 6: CAL SOURC2 TEMP = % (90 - 94 %)

Row 7: (blank)

Row 8: > END OF VERIFICATION <

Row 9: (blank)

Row 10: (blank)

PKD APPLICATION DEFINITION

Frame number (I5) __187____

Row 1: CHECK-OUT COMPLETED SUCCESSFULLY

Row 2: MODE SWITCHED TO STANDBY

Row 3: SELECT NEXT ACTION

Row 4: TIME IS 72.48

Row 5: NEW EXP NEXT

Row 6: PHASE CHECKOUT

Row 7: NEXT REQUIRED ACTION IS

Row 8: BEGIN OPS AT 73.050

Row 9: XT ACTIO

Row 10: 73.003

120

PKD APPLICATION DEFINITION

Frame number (I5) ___186___

1	C	H	E	C	K	-	O	U	T		I	N	D	I	C	A	T	E	S		F	A	U	L	T		S	H	O	W	N
2	P	R	E	S	S		S	W	I	T	C	H		F	O	R		C	O	N	T	I	N	G	E	N	C	Y		A	C	T
3			C	R				R	E	P	L	A	N		D	O		N	E	X	T									
4								E	X	P	E	R	.		C	H	E	C	K	O	U	T								
5		R	E	P	L	A	C	E																					
6	S	O	U	R	C	E	1		S	O	U	R	C	E	2		S	O	U	R	C	E	3		S	O	U	R	C	E	5
7		R	E	A	L	I	G	N		R	E	P	L	A	C	E		R	E	P	A	I	R							
8		A	R	R	A	Y			C	O	O	L	E	R			E	L	E	C	T	.								
9																													
10																													

PKD APPLICATION DEFINITION

Frame number (I5) ___189___

1	C	H	E	C	K		L	I	S	T		7	0		R	E	P	L	A	C	E		C	A	L		S	O	U	R	C
2		P	R	E	S	S		S	W	I	T	C	H		W	H	E	N		C	O	M	P	L	E	T	E	D		
3	A	.	P	O	W	E	R		D	.	C	H	A	N	G	E	G	.	P	O	W	E	R						
4			O	F	F				S	O	U	R	C	E				U	P									
5	B	.	L	O	O	S	E		E	.	R	E	S	E	T		T	I	M	E		R	F	G		T	I	M	E		I	S
6	4		S	C	R	E	W	S			C	O	V	E	R		0	.	5	0		O	R	B			7	2	.	4	9
7	C	.	R	E	M	O	V	E	F	.	T	I	G	H	T										D	E	A	D	L	I	N	E
8			C	O	V	E	R		4		S	C	R	E	W	S										7	3	.	0	5
9																													
10																													

121

APPENDIX E
WORD RECOGNITION SYSTEM USER'S GUIDE
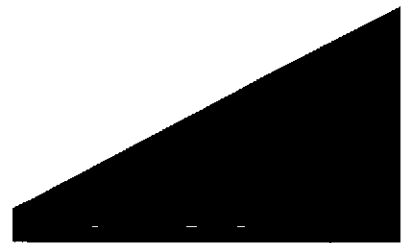
Appendix E
WORD RECOGNITION SYSTEM USER'S GUIDE

The User's Guide is presented in an earlier work, Design and Performance of a Large Vocabulary Discrete Word Recognition System, Volume II, published by McDonnell Douglas Astronautics Company, Report No. MDC G4829, November 1973.